ADAPTIVE NOISE CANCELLERS : THEIR DESIGN AND IMPLEMENTATION

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By
L. DEIVASIGAMANI

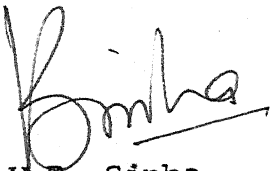
to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1981

CERTIFICATE

This is to certify that this thesis, entitled: "ADAPTIVE NOISE CANCELLERS, THEIR DESIGN AND IMPLEMENTATION," is a record of the work carried out by Mr. L. Deivasigamani under my supervision and that this has not been submitted elsewhere for a degree.



V.P. Sinha
Department of Electrical Engineering
Indian Institute of Technology
Kanpur

January, 1981

ACKNOWLEDGEMENTS

I wish to record my profound gratitude to Professor V.P. Sinha for the freedom he gave me in the choice of this subject and for his constant guidance and encouragement throughout the course of this work.

I would like to express my sincere gratitude to Mr. K.G. Narayanan, ADE, Ministry of Defence, for the useful discussion I had with him that helped me choose this topic.

Thanks are due also to Lt. S. Ramaprasad, Indian Navy, and Dr. R.V. Chalam for assisting me in proof-reading and compiling the thesis.

I am greatly indebted to ADE for providing me the opportunity to do the post-graduate studies at IIT/K.

Finally, I extend my appreciation to Mr. R.N. Srivastava who meticulously typed the manuscript.

January, 1981 L. Deivasigamani

CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

B	Bandwidth
d	Primary input
E	Expected value
i	Tap index
j	Discrete-time index
k	Discrete-frequency index
m	Block index
M	Misadjustment
n	Noise
N	Number of taps
R	Correlation function
s	Signal
S	Power spectral density
T	Delay; Sampling interval; Transpose
w	Tap variable
x	Reference input
y	Adaptive filter output
z	z-transform variable; ANC output
μ	Adaptation constant
τ	Time constant; Rate of adaptation
ξ	Mean-square error
Δ	Gradient
λ	Eigen value

ρ	Signal-to-noise density ratio; Correlation coefficient
$\sum$	Summation
*	Convolution operation; Optimum value; Conjugate
Caret	Estimate

SECTION 1

INTRODUCTION

Filters are devices that process incoming signals so as to eliminate noise, carry out smoothing, identify signals as belonging to different classes, or predict the next input signal based on present and past inputs. Filters may be divided into two classes: fixed and adaptive. The impulse responses of fixed filters do not vary with time, whereas that of adaptive filters are, in general, time-varying.

The design of fixed filters is based upon prior knowledge about signals and noise. Adaptive filters, on the other hand, have the ability to adjust their own parameters, and their design does not require prior knowledge of signal and noise characteristics. These are, in essence, a class of "learning machines" whose parameters are self-adjustable in accordance with estimated or measured statistical characteristics of input and output signals.

The classical theories of detection and estimation are usually presented in a fixed or non-adaptive context. In such a context it is assumed that some statistical properties of signal and noise are known beforehand. In the Wiener theory of estimation, for instance, one needs to know the second-order properties, viz. the correlation functions of signals. Similarly, in detection theory, the optimal receiver

makes use of a likelihood ratio which requires signal and noise probability distributions.

In practice a priori knowledge of signal and noise characteristics may not readily be available. Moreover, if the signal and noise parameters change, as is usually the case, the original fixed optimum receiver will no more be optimum, resulting in degradation of performance. Adaptive filters have found extensive use in such applications [1-1]. These filters "learn" the characteristics of the signals initially and track them, if they vary slowly enough, by adjusting their impulse responses. Adjustment is accomplished through an algorithm that responds to a suitably derived error signal. Thus, with a proper algorithm the filter can operate under changing conditions and can readjust itself continually to minimize the error signal. Adaptive filters have been used in adaptive smoothing, filtering, equalization, prediction, and estimation of signals, and identification of system parameters; they have also been used in spectral estimation, in noise whitening and in speech signal processing [1-2, 1-3].

A particular application of adaptive filters is in "noise cancelling". It is an alternative method of estimating a signal corrupted by additive noise. A "reference input" is derived from one or more sensors located in the noise field at points where the signal is weak or undetectable. The reference input is adaptively filtered and subtracted from a "primary input" consisting of the signal and the additive

noise. This results in an output that is the best estimate, in the mean-square sense, of the actual or desired signal. Adaptive filtering before subtraction allows processing of signals that are deterministic or stochastic, stationary or nonstationary.

This thesis deals with the theory of adaptive noise cancellers, their design, and their implementation by means of microprocessors. In Section 2, the principles of noise cancelling are reviewed and the various design rules available at present for adaptation based on the so-called LMS algorithms are brought within one framework. Then the relative merits of some recent versions of the LMS algorithm are discussed, and new computationally efficient algorithms are proposed.

After discussing the advantages and limitations in Section 3, the noise cancelling technique is illustrated in Section 4 by considering various applications and presenting simulation results. In Section 5, the possibility of using microprocessors for implementing noise cancellers is examined. The hardware and software aspects of implementing a typical noise canceller are discussed; also, a real-time noise canceller scheme is proposed. The thesis concludes in Section 6 by summarizing the results and with suggestions for further work.

SECTION 2

PRINCIPLES AND ALGORITHMS

2.1 The Concept

The concept of adaptive noise cancelling may be explained with the aid of Figure 2.1. A signal s, transmitted over a channel and corrupted by an additive noise or interference n_d, forms the primary input to the adaptive noise canceller (ANC). A noise n_x, derived from the same source field as the signal, forms the reference input, the sensor for n_x being so placed as to pick up only the noise components. The reference noise is uncorrelated with the signal, but correlated in some unknown way with the primary noise.

The noise n_x is filtered by means of an adaptive filter to produce an output y that is a close replica of n_d. The output of the filter is then subtracted from the primary input to produce the system output, z. The objective is to produce a system output that is the best fit, in the mean-square sense, to the signal s. In the figure, the feedback from the system output to the adaptive filter is meant to provide requisite parameter adjustments. It is assumed that s, n_d, n_x and y are statistically stationary and have zero means.

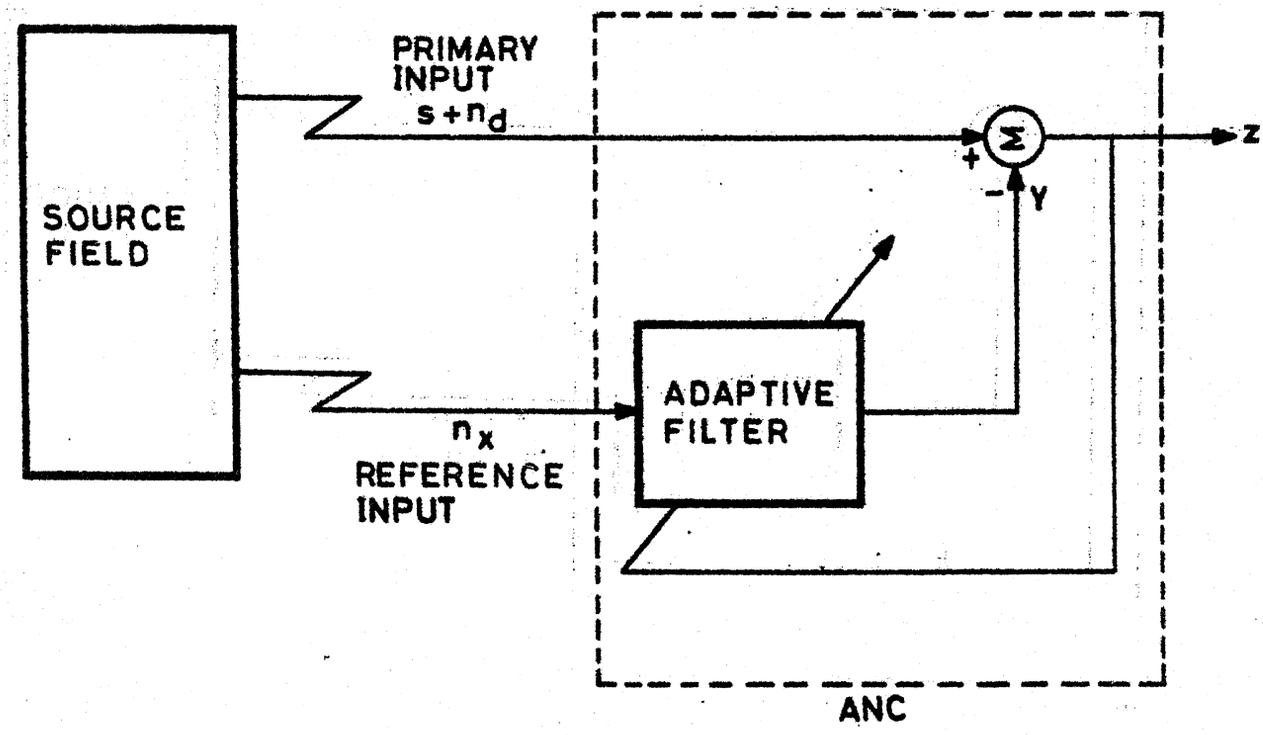


Fig. 2.1: Noise cancelling with an adaptive filter.

Referring to Figure 2.1, the system output is

$$z = s + n_d - y \ .$$

The mean-square value of the output is

$$E[z^2] = E[s^2] + E[(n_d-y)^2] + 2E[s(n_d-y)] \ .$$

If n_x is uncorrelated with s, then y and hence n_d-y will also be uncorrelated with s. Accordingly,

$$E[z^2] = E[s^2] + E[(n_d-y)^2] \ . \qquad (2.1)$$

The mean-square error (mse), e, between the output and the desired signal is

$$E[e^2] = E[(z-s)^2]$$

$$= E[(n_d-y)^2] \ . \qquad (2.2)$$

Inasmuch as the signal is unaffected by the adaptive filter, it is seen from Eqs. (2.1) and (2.2) that minimizing the mse is equivalent to minimizing the mean-square value of the system output. When $E[z^2]$ is minimum, $E[(n_d-y)^2] = 0$. Therefore, $y = n_d$ and $z = s$. Hence, minimizing the mean-square value of the output causes the signal to be perfectly noise free.

When the reference input is completely uncorrelated with the primary input,

$$E[z^2] = E[(s + n_d)^2] + 2E[-y(s + n_d)] + E[y^2]$$

$$= E[(s + n_d)^2] + E[y^2] .$$

In this case, the mse is minimized if $E[y^2]$ is minimized, which is accomplished by the filter by making its output zero.

The adaptive filter, indeed, so adjusts its parameters as to have, ideally, the inverse of the channel characteristics, thus reproducing the correlated noise components that originated at the source field and were degraded by the channel. It may be noted that, though we have considered only the case of stochastic signals, the discussion is equally valid for the case of deterministic signals, as will be seen in Section 4.

It is now explained how the minimization of the mean-square output(i.e., mse) is actually carried out. For the sake of discussion, the signals are assumed ergodic.

2.2 The Wiener Solution

Let the jth sample of the primary input be denoted by d(j). The jth set of reference samples may be represented by a random vector $\underline{X}(j)$, where

$$\underline{X}^T(j) = [x_o(j), x_1(j), \ldots, x_{n-1}(j)] .$$

A noise estimate, y(j), may be obtained as a linear combination of the reference samples, $\underline{X}(j)$; i.e.,

$$y(j) = \underline{W}^T \underline{X}(j) = \underline{X}^T(j)\, \underline{W}$$

where the components of the vector

$$\underline{W}^T = [\, w_o,\ w_1,\ \ldots,\ w_{n-1} \,]$$

are the weights whose values have to be determined in accordance with a specific criterion of performance. If only the second-order properties of the signals are known, the simplest possible criterion is the minimization of mse, which is accomplished by minimizing the output power. The corresponding weight vector $\underline{W}$ is then determined with the help of the Wiener filter theory.

The output at the jth instant is given by

$$z(j) = d(j) - \underline{W}^T \underline{X}(j) \ . \tag{2.3}$$

Squaring Eq. (2.3) and taking expected value of both sides yield

$$E[z^2(j)] = E[d^2(j)] - 2E[d(j)\, \underline{X}^T(j)]\, \underline{W} + \underline{W}^T E[\underline{X}(j)\, \underline{X}^T(j)]\, \underline{W} \ . \tag{2.4}$$

Let

$$\underline{R}_{dx} \triangleq E[d(j)\, \underline{X}(j)] \tag{2.5}$$

$$= E \begin{bmatrix} d(j)\, x_o(j) \\ d(j)\, x_1(j) \\ \vdots \\ d(j)\, x_{n-1}(j) \end{bmatrix}$$

where $\underline{R}_{dx}$ denotes the vector of cross-correlations between the reference samples and the primary input, and

$$R_x \triangleq E[\underline{X}(j)\,\underline{X}^T(j)] \tag{2.6}$$

$$= E\begin{bmatrix} x_o^2(j) & x_o(j)\,x_1(j) & \cdots & \\ x_1(j)\,x_o(j) & x_1^2(j) & \cdots & \\ \vdots & & & \vdots \\ & & \cdots & x_{n-1}^2(j) \end{bmatrix}$$

where R_x is the covariance matrix of the reference samples. Now, the mean-square output may be expressed as

$$E[z^2(j)] = E[d^2(j)] - 2\underline{R}_{dx}^T\,\underline{W} + \underline{W}^T R_x\,\underline{W} \; . \tag{2.7}$$

Following the usual procedure, the condition for minimum output power is found by differentiating Eq. (2.7) with respect to the weight vector and equating it to zero. Accordingly, the gradient is given by

$$\nabla E[z^2(j)] = -2\underline{R}_{dx} + 2R_x\,\underline{W}$$

where

$$\nabla E[z^2(j)] = \begin{bmatrix} \frac{\partial E[z^2(j)]}{\partial w_0} \\ \frac{\partial E[z^2(j)]}{\partial w_1} \\ \vdots \\ \frac{\partial E[z^2(j)]}{\partial w_{n-1}} \end{bmatrix} . \qquad (2.8)$$

The optimal weight vector, $\underline{W}_*$, is obtained by equating the gradient to zero. Therefore,

$$R_x \underline{W}_* = \underline{R}_{dx} .$$

The matrix R_x, which is Toeplitz, is generally positive definite and, therefore, invertible. Accordingly, the above relation may be written as

$$\underline{W}_* = R_x^{-1} \underline{R}_{dx} \qquad (2.8)$$

which is the Wiener-Hopf equation in matrix form.

In practice a direct solution for the weight vector is difficult to obtain, for an nxn matrix needs to be inverted and $n(n + 1)/2$ autocorrelation and cross-correlation measurements have to be made to obtain the elements of R_x and $\underline{R}_{dx}$. This evidently would entail a huge computational burden as well as storage of large amounts of data. Furthermore, this procedure needs to be repeated if the signal statistics change. A number of algorithms

have therefore been proposed which compute the Wiener solution iteratively based on the incoming data samples. Among them is the so-called LMS algorithm, which is simple from an implementation point of view, yet effective in many noise cancelling problems [2-1]. This algorithm is chosen for discussion here both to gain an insight into the behaviour of ANC's and to obtain simplified design rules. Some other, recent algorithms, which are of practical interest, are also discussed and compared with the LMS algorithm towards the end of this section.

2.3 The LMS Algorithm

Equation (2.7) indicates that, for stationary input signals, the mse is a second-order function of the weight vector and, therefore, is a convex function, the minimum of which will correspond to the optimum weight values.

To find this minimum one of the gradient methods, namely the steepest descend method, is usually employed. The method is perhaps best explained by considering a one-dimensional case shown in Figure 2.2. Suppose that $w(j)$, the weight value at the jth instant, has some value different from w_*. We find the slope of the curve at this point. If the slope is negative, $w(j)$ is to the left of the minimum, and if it is positive, $w(j)$ is to the right. Therefore, if at the next time instant $w(j)$ is chosen as

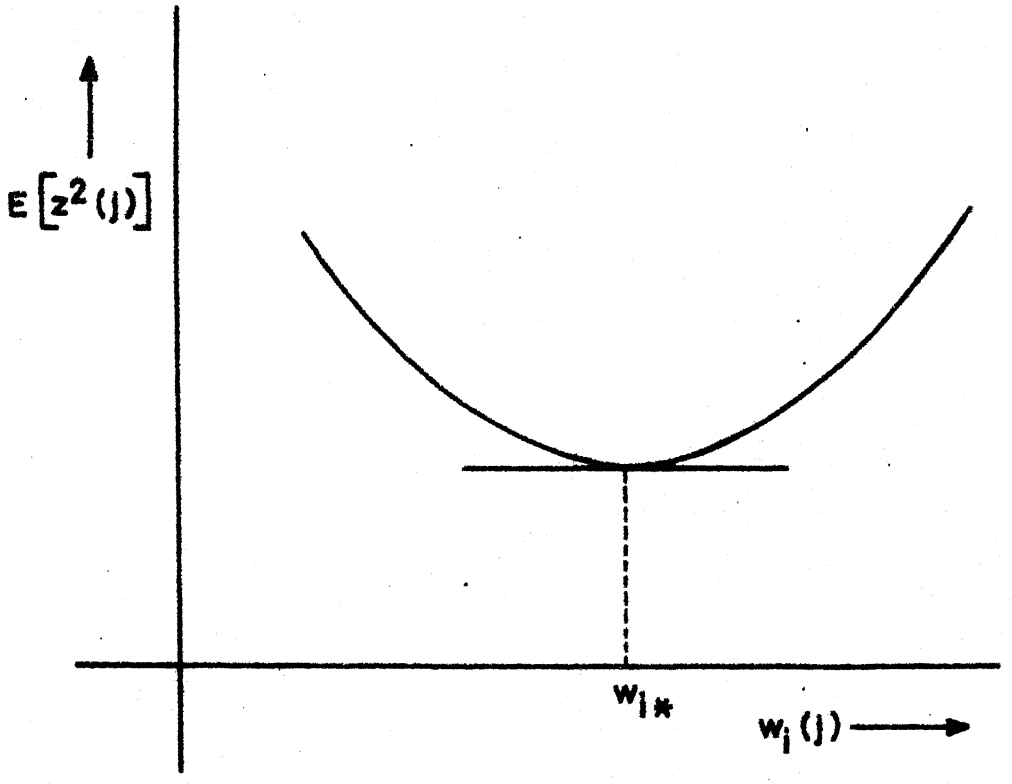


Fig. 2.2: Variation of mse with ith filter weight.

$$w(j+1) = w(j) - \mu \left.\frac{dE[z^2]}{dw}\right|_{w = w(j)}$$

where μ is a constant, then $w(j+1)$ is shifted in the right direction to reduce the mse.

For the n-dimensional case, where the gradient $\nabla E[z^2(j)]$ rather than the slope is of interest, the updating algorithm becomes

$$\underline{W}(j+1) = \underline{W}(j) - \mu \nabla E[z^2(j)]. \qquad (2.9)$$

Equation (2.9) implies simultaneous adjustment of all the filter weights. The adjustment of each of the filter weights is proportional to the corresponding component of the gradient vector. Interchanging the operations of expectation and differentiation yields

$$w(j+1) = w(j) - 2\mu \nabla E[z(j) \nabla z(j)].$$

From Eq. (2.3), we obtain

$$\nabla z(j) = -\underline{X}(j)$$

Therefore,

$$\underline{W}(j+1) = \underline{W}(j) + 2\mu E[z(j) \underline{X}(j)], \qquad (2.10)$$

which shows that the incremental weight update is a cross-correlation between the error sequence and the input data sequence, when the sequences are ergodic. Thus the

process of optimization can be pictured as the process of reaching the point in time when the error sequence is orthogonal to the data sequence.

In implementing Eq. (2.10), the evaluation of the cross-correlation terms requires a large amount of computation as well as storage. A particular form of the above algorithm that overcomes this difficulty has been introduced by Widrow and Hopf [2-1]. The algorithm makes the approximation that the gradient of a single time sample of the squared output is an estimate of the gradient of the mean square function; i.e.,

$$\nabla E[z^2(j)] \simeq \hat{\nabla}[z^2(j)] \; . \tag{2.11}$$

Such an estimate is unbiased; i.e.,

$$E\{\hat{\nabla}[z^2(j)]\} = \nabla E[z^2(j)] \; .$$

Equation (2.9) now becomes

$$\underline{W}(j+1) = \underline{W}(j) + 2\mu z(j)\,\underline{X}(j) \tag{2.12}$$

where

$$z(j) = d(j) - \underline{W}^T(j)\,\underline{X}(j) \; .$$

The above algorithm means that the next weight vector is obtained by adding to the present weight vector the input vector scaled by the present value of the output. It is simple and easy to implement. Although it makes use of

gradients of mse functions, it does not require squaring, averaging or differentiation. It is commonly referred to as the LMS algorithm. The constant μ is known as the adaptation constant or step size. The realization of the above algorithm by means of an adaptive filter is shown in Figure 2.3. The components of $\underline{X}(j)$ are obtained as a delayed sequence of the reference samples.

By analyzing the LMS algorithm, expressions for rate of adaptation and deviation from the least mse after convergence can be obtained. The analysis may be done either in the sample domain or in the frequency domain. We adopt the sample domain approach.

2.3.1 Rate of Adaptation and Learning Curve

Taking expected value of both sides of Eq. (2.12) yields

$$E[\underline{W}(j+1)] = E[\underline{W}(j)] + 2\mu E[d(j)\,\underline{X}(j)] - 2\mu E[\underline{X}(j)\,\underline{X}^T(j)\,\underline{W}(j)] . \qquad (2.13)$$

It is assumed that the time between successive iterations is sufficiently long so that $\underline{X}(j)$ and $\underline{X}(j+1)$ are uncorrelated. The weight vector $\underline{W}(j)$ is then uncorrelated with $\underline{X}(j)$. Equation (2.13), therefore, takes the form

$$E[\underline{W}(j+1)] = E[\underline{W}(j)] + 2\mu \underline{R}_{dx} - 2\mu R_x E[\underline{W}(j)]$$

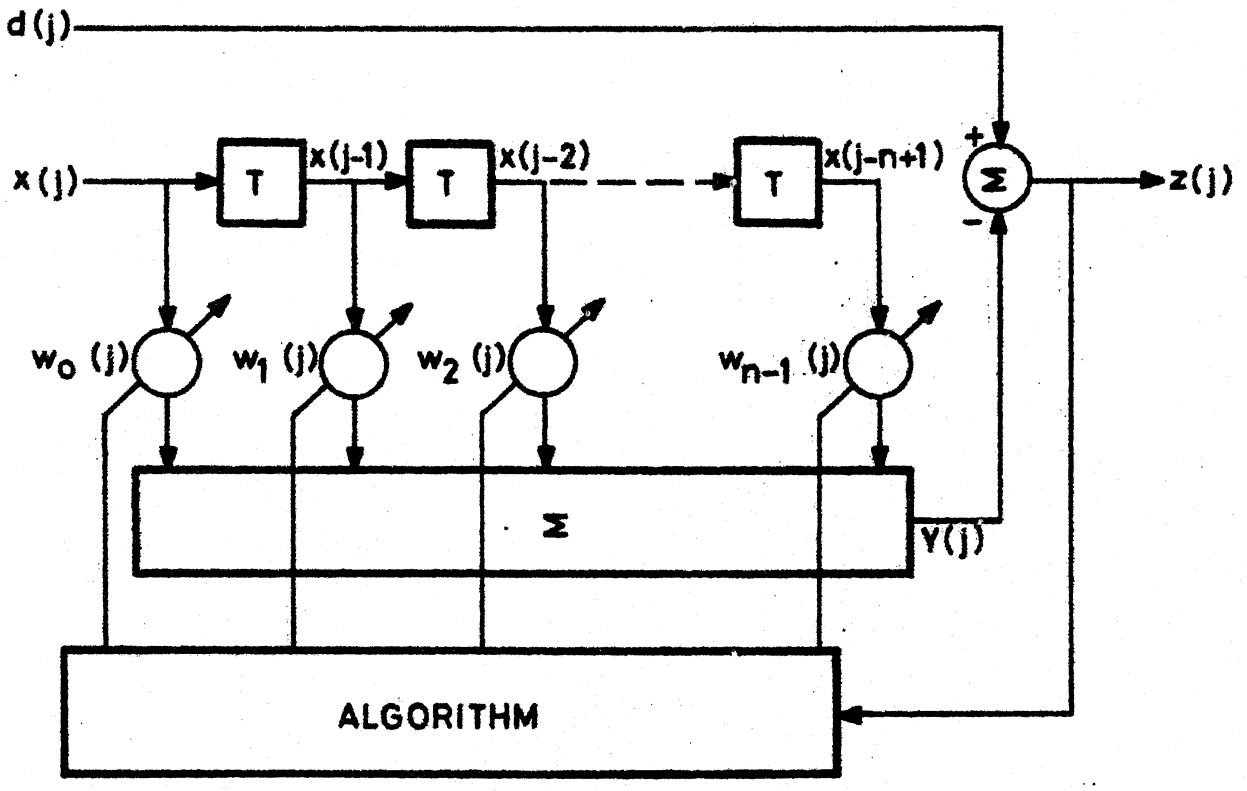


Fig. 2.3: Block diagram of the adaptive noise canceller.

i.e.

$$= [I - 2\mu R_x] E[\underline{W}(j)] + 2\mu \underline{R}_{dx} .$$

With an initial vector $\underline{W}(o)$, the above expression may be iterated j times to obtain

$$E[\underline{W}(j+1)] = (I - 2\mu R_x)^{j+1} \underline{W}(o) + 2\mu \sum_{m=o}^{j} (I - 2\mu R_x)^m \underline{R}_{dx} . \qquad (2.14)$$

Conventional analysis transforms Eq. (2.14) to an orthogonal coordinate system where the covariance matrix R_x is diagonalized; the result is that each weight can be expressed independent of the other weights. Expressing R_x as

$$R_x = Q^{-1} \Lambda Q = Q^T \Lambda Q \qquad (2.15)$$

where Λ is the diagonal matrix of eigenvalues, and Q, the orthonormal matrix of eigenvectors, we obtain from Eqs. (2.14) and (2.15) after simplification,

$$QE[\underline{W}(j+1)] = (I - 2\mu\Lambda)^{j+1} Q \underline{W}(o) + 2\mu \sum_{m=o}^{j} (I - 2\mu\Lambda)^m Q \underline{R}_{dx} . \qquad (2.16)$$

Let $j \to \infty$. Consider now the term

$$\sum_{m=o}^{\infty} (I - 2\mu\Lambda)^m . \qquad (2.17)$$

Since each diagonal element of the matrix $(I - 2\mu\Lambda)$ forms a geometric series, the ith element, λ_i, may be expressed, using the standard summation formula, as

$$\sum_{m=0}^{\infty} (1 - 2\mu\lambda_i)^m = \frac{1}{2\mu\lambda_i}$$

provided that

$$|1 - 2\mu\lambda_i| < 1 .$$

Similarly,

$$\sum_{m=0}^{\infty} (I - 2\mu\Lambda)^m = \frac{1}{2\mu}\Lambda^{-1} \qquad (2.18)$$

so long as

$$|1 - 2\mu\lambda_{max}| < 1$$

i.e.,

$$0 < \mu < \frac{1}{\lambda_{max}} \qquad (2.19)$$

where λ_{max} is the maximum eigenvalue of R_x. Similarly, if the inequality (2.19) is satisfied, the RHS first term of Eq. (2.16) equals zero as $j \to \infty$. Therefore,

$$\lim_{j\to\infty} E[\underline{W}(j+1)] = R_x^{-1}\underline{R}_{dx} .$$

Thus, as the number of iterations increases, the mean weight vector converges to the Wiener solution. The equation may be interpreted as follows: If we consider an

ensemble of adaptive processes, each having individual stationary ergodic inputs drawn from the same ergodic process, with all initial weight vectors equal, the weight vector, obtained by averaging over the ensemble at iteration j, will converge to the Wiener solution, in the limit.

In practice it is difficult to know the value of λ_{max} to set the bound on μ. However, a sufficient condition for convergence may be obtained by noting that

$$\lambda_{max} \leq t_r R_x$$

where $t_r R_x$, the trace of R_x, corresponds to the reference input power, which is generally known. Therefore, a sufficient condition to ensure convergence of the weight vector is

$$0 < \mu < \frac{1}{t_r R_x} . \qquad (2.20)$$

The time constant, τ_i, of the ith weight is defined as the time taken for the weight to get to $1-e^{-1}$ of its steady-state value for zero initial conditions and step inputs. Here, the unit of time is the time taken for one iteration cycle. For the ith weight, its value at the jth instant is the sum of j terms of the geometric series (2.16) with geometric ratio $1-2\mu\lambda_i$. Therefore,

$$1 - (1 - 2\mu\lambda_i)^j \Big|_{j = \tau_i} = 1 - e^{-1}$$

or, equivalently,

$$\tau_i = \frac{-1}{\ln (1 - 2\mu \lambda_i)} .$$

In most practical situations $|2\mu \lambda_i| << 1$, in which case,

$$\ln (1 - 2\mu \lambda_i) \simeq - 2\mu \lambda_i$$

i.e.,

$$\tau_i \simeq \frac{1}{2\mu \lambda_i} ; \quad i = 0, 1, \ldots n-1. \tag{2.21}$$

Hence the transients consist of a linear combination of exponentials with time constants given by Eq. (2.21).

The progress of an adaptive process can be monitored by displaying its "learning curve". It is a plot of the expected value of the mse at each stage of the learning process as a function of the number of adaptation cycles. The underlying transient phenomenon that takes place in the weight values is exponential. Therefore the mse, which is a quadratic function of the weights, has also an exponential decay. The time constant of the learning curve is determined as follows:

Combining Eqs. (2.7) and (2.8) yields

$$E [z^2(j)]_{min} = E [d^2(j)] - \underline{W}_*^T \underline{R}_{dx} . \tag{2.22}$$

From Eqs. (2.7), (2.8) and (2.22), we obtain

$$E[z^2(j)] = E[z^2(j)]_{min} + (\underline{W}(j) - \underline{W}_*)^T R_x (\underline{W}(j) - \underline{W}_*) . \tag{2.23}$$

It is seen that the mse is quadratic in the weight vector and, hence, has a geometric ratio, $(1 - 2\mu\lambda_i)^2$. The time constants of the learning curve are, therefore,

$$\tau_{i_{mse}} \simeq \frac{1}{4\mu\lambda_i} ; \qquad i = 0, 1, \ldots n-1. \tag{2.24}$$

If the eigenvalues do not differ widely, it is reasonable to express the time constants of the learning curve as a function of the average eigenvalue, λ_{av}, where

$$\lambda_{av} = \frac{1}{n} \sum_{i=0}^{n-1} \lambda_i$$

$$= \frac{t_r R_x}{n} .$$

Accordingly, Eq. (2.24) takes the form

$$\tau_{mse} \simeq \frac{n}{4\mu\, t_r R_x} , \tag{2.25}$$

which gives a formula for the average or effective time constant of the learning curve that is useful even when the eigenvalues are highly disparate. This time constant is referred to as the "rate of adaptation" in the literature (A more appropriate term might be adaptation time).

When actual experimental learning curves are plotted, they are generally of the form of noisy exponentials because of the noise due to statistical averages taken with limited sample sizes. This point is discussed next.

2.3.2 Maladjustment with LMS Adaptation

It is of interest to know the penalty incurred in making the approximation of (2.11). Whereas the mean-weight vector, under appropriate conditions, converges to its proper value in the limit, its variance often induces a certain amount of degradation. This degradation in performance is measured in terms of a dimensionless quantity, known as "misadjustment" [2-2]. It is defined as the ratio of the average excess mse to the minimum mse. An expression for the misadjustment may be obtained under the steady-state conditions as follows:

As before, it is assumed that the successive data vectors $\underline{X}$'s are uncorrelated over time, and that $\underline{W}(j) \simeq W_*$. The true gradient $\nabla E[z^2(j)]$ may be expressed as a sum of the gradient estimate $\hat{\nabla}[z^2(j)]$ and a zero-mean gradient estimate noise vector, $\underline{N}(j)$; i.e.,

$$\nabla \xi(j) = \hat{\nabla}[z^2(j)] + \underline{N}(j) \tag{2.26}$$

where $\xi(j) \triangleq E[z^2(j)]$. Substituting for $\hat{\nabla}$ yields

$$\underline{N}(j) = -2z(j)\,\underline{X}(j) + \nabla\xi(j).$$

Also, when $\underline{W}(j) = \underline{W}_*$, $\nabla\xi(j) = 0$. Therefore,

$$\underline{N}(j) = -2z(j)\,\underline{X}(j). \tag{2.27}$$

The covariance matrix of $\underline{N}(j)$ is

$$R_n(j) \triangleq E[\underline{N}(j)\,\underline{N}^T(j)]$$

$$= 4E[z^2(j)\,\underline{X}(j)\,\underline{X}^T(j)].$$

In accordance with the orthogonality theorem, when $\underline{W}(j) = \underline{W}_*$, $z(j)$ and $\underline{X}(j)$ are uncorrelated. If they are assumed zero-mean Gaussian, then $z(j)$ and $\underline{X}(j)$ are statistically independent. Therefore,

$$R_n(j) = 4E[z^2(j)]\,R_X$$

or, in the steady-state,

$$= 4\,\xi_{min}\,R_X\,. \tag{2.28}$$

Since $\underline{X}$'s are stationary and uncorrelated over time, it follows that the gradient noise vector $\underline{N}(j)$ is also stationary and uncorrelated over time. Equation (2.28) may be rewritten as

$$R'_n = 4\,\xi_{min}\,\Lambda \tag{2.29}$$

where

$$\underline{N}'(j) = Q\,\underline{N}(j)\,.$$

Such a transformation makes the analysis simpler.

The effect of gradient noise on the steady-state weight vector can now be determined. The LMS algorithm with a noisy gradient estimate can be expressed as

$$\underline{W}(j+1) = \underline{W}(j) - \mu \hat{\nabla}[z^2(j)]$$

$$= \underline{W}(j) - \mu [\nabla\xi(j) - \underline{N}(j)] \quad . \tag{2.30}$$

Equation (2.23) may be reexpressed as

$$\xi(j) = \xi_{min} + \underline{V}^T(j) R_x \underline{V}(j) \tag{2.31}$$

where

$$\underline{V}(j) \triangleq \underline{W}(j) - \underline{W}_*. \tag{2.32}$$

An alternative expression for the true gradient , obtained from Eq. (2.31), is

$$\nabla\xi(j) = 2R_x \underline{V}(j) \quad . \tag{2.33}$$

From Eqs. (2.30), (2.32) and (2.33), we obtain

$$\underline{V}(j+1) = \underline{V}(j) - \mu [(2R_x\underline{V}(j) + \underline{N}(j)]$$

or, letting $\underline{V}'(j) = Q \underline{V}(j)$,

$$\underline{V}'(j+1) = \underline{V}'(j) - 2\mu\Lambda \underline{V}'(j) + \mu \underline{N}'(j)$$

$$= (I - 2\mu\Lambda) \underline{V}'(j) + \mu \underline{N}'(j) \quad . \tag{2.34}$$

Since $\underline{N}(j)$ is uncorrelated over time, $\underline{N}'(j)$ and $\underline{V}'(j)$ are also uncorrelated. Postmultiplying both sides of Eq. (2.34) by their transposes, and taking expected value of both sides yield

$$R'_v(j+1) \triangleq E[\underline{V}'(j+1)\,\underline{V}'^{T}(j+1)]$$

$$= (I - 2\mu\Lambda)\,R'_v(j)(I - 2\mu\Lambda)$$

$$+ \mu^2 R'_n \quad . \tag{2.35}$$

The covariance matrix $R'_v(j)$ is diagonal. If $\underline{V}'(j)$ is assumed to be stationary, Eq. (2.35) may be rewritten as

$$R'_v = (I - 2\mu\Lambda)^2 R'_v + \mu^2 R'_n$$

Substituting for R'_n from Eq. (2.29) and rearranging the terms yield

$$(I - \mu\Lambda)\,R'_v = \mu\,\xi_{min}\,I \quad .$$

Assuming that $\mu << 1$, which is consistent with the assumption that $\underline{W}(j) \simeq \underline{W}_*$, it is seen that

$$\mu\Lambda << I \quad .$$

Therefore,

$$R'_v \simeq \mu\,\xi_{min}\,I$$

i.e.,

$$R_v = \mu\,\xi_{min}\,I \quad . \tag{2.36}$$

The above expression shows that, under the stated assumptions, the components of the weight-vector noise are of equal variance and are mutually uncorrelated.

Due to the random noise, caused by the gradient estimate noise, the weight vector is on the average misadjusted from its optimal setting. This results in an excess mse. From Eq. (2.31), an expression for the excess mse may be obtained as

$$\xi_{exs} \triangleq \underline{V}'^{T}(j)\,\Lambda\,\underline{V}'(j) .$$

The average excess mse is

$$E[\xi_{exs}] = \sum_{i=0}^{n-1} \lambda_i E[\{v_i'(j)\}^2] . \qquad (2.37)$$

From Eqs. (2.36) and (2.37) we obtain

$$E[\xi_{exs}] = \mu \xi_{min} \sum_{i=0}^{n-1} \lambda_i$$

$$= \mu \xi_{min} t_r R_x .$$

Hence the expression for the misadjustment is, by definition,

$$M = \frac{E[\xi_{exs}]}{\xi_{min}}$$

i.e.

$$= \mu t_r R_x . \qquad (2.38)$$

When the eigenvalues are sufficiently similar, an approximate expression for the misadjustment in terms of τ_{mse}, obtained from Eq. (2.25), is

$$M \simeq \frac{n}{4\,\tau_{mse}} \,. \qquad (2.39)$$

Equations (2.25) and (2.39) provide approximate expressions for the rate of adaptation and the deviation from the minimum mse, respectively. Now the question arises as to how to determine the number of taps. The delay between the taps must be chosen in accordance with the sampling theorem. Let B Hz be the bandwidth of the reference signal. The delay T is chosen as

$$T \leq \frac{1}{2B} \,.$$

If ΔB is the frequency resolution of the adaptive filter, then the total real-time length of the delay line is

$$nT = \frac{1}{\Delta B} \,.$$

Thus,

$$n \geq \frac{2B}{\Delta B} \,. \qquad (2.40)$$

The expressions (2.20), (2.25), (2.39) and (2.40) provide a set of approximate formulae for designing ANC's with adaptation based on the LMS algorithm. The first three equations depend on the reference input power, which could be measured approximately by a time average power of the reference samples. In practice some trial and error procedure is necessary to achieve desired results.

A brief discussion about the overall performance of the ANC is now in order. In broad-band signal processing increasing the number of taps may improve the performance, for it permits the adjustment of gain and phase at many frequencies simultaneously. Nevertheless, for a fixed rate of adaptation a larger numbers of taps increase misadjustment as is evident from Eq. (2.39). Similarly, increasing the adaptation constant speeds up the adaptive process, but it increases the misadjustment, too. Alternatively, for a fixed n, more the reference input power the faster is the adaptation as seen from Eq. (2.25). There is a tradeoff possible between the misadjustment and the rate of adaptation. It has been pointed out [2-2] that the rate of adaptation is optimized when the excess mse resulting from adapting too rapidly equals twice the excess mse resulting from adapting too slowly. When a slow, but precise adaptation is desired μ is chosen such that

$$0 < \mu << \frac{1}{t_r R_x} . \qquad (2.41)$$

Although the assumption of independent successive samples is invoked in the convergence proof for the mean weight vector, Daniel [2-3] has shown that adaptation using highly correlated successive samples does converge to the Wiener solution. However, it leads to slightly higher steady-state mse than does adaptation using statistically independent successive samples. Also, Shensa [2-4] has

pointed out that the Wiener solution is a good approximation for the steady-state weight vector as long as it is larger than the misadjustment; but should it be zero, one may not, in general, ignore the misadjustment no matter how small. In a stationary environment, it is possible to decrease μ [1-1] as the adaptation progresses bringing the gradient eventually to zero and so also the misadjustment.

There are indeed a host of other algorithms which might in practice prove useful. Some of the recent algorithms are discussed and compared in what follows.

2.4 The Modified LMS Algorithm (MLMS)

The LMS algorithm is conditionally stable with the result that the rate of adaptation and, in general, the performance of the filter are constrained. A modification [2-5] has been introduced in the LMS algorithm that removes the constraint on μ and still achieves the same mean steady-state weight as the LMS algorithm.

The recursive relation (2.12) is modified as

$$\underline{W}(j) = \underline{W}(j-1) + 2\mu\, z(j)\, \underline{X}(j) \qquad (2.42)$$

$$z(j) = d(j) - \underline{W}^{T}(j)\, \underline{X}(j) .$$

That is, the present weight $\underline{W}(j)$ is obtained from the present values of $z(j)$ and $\underline{X}(j)$ in contrast to the LMS algorithm which generates the next weight from the present values of $z(j)$ and $\underline{X}(j)$. The condition for stability of the

one-dimensional case of the algorithm is given by [2-5]

$$\frac{1}{1 + 2\mu |x|^2} < 1$$

which is always satisfied if μ and $|x|^2$ are non-zero.

Although the experimental performance of this algorithm has been reported to be better than the LMS algorithm, its implementation is limited by the computation involved; the authors [2-5] have suggested that one way is to compute each weight in a closed-loop fashion while the other weights are frozen; i.e.,

$$w_i(j) = w_i(j-1) + 2\mu\, z(j)\, x(j-i)$$

$$z(j) = d(j) - w_i(j)\, x(j-i) - \sum_{m=1,\ m\neq i}^{n} w_m(j-1)\, x_m(j) .$$

The actual output signal resulting from this procedure is then taken to be

$$z(j) = d(j) - \sum_{i=1}^{n} w_i(j)\, x(j-i) .$$

An alternative and perhaps more elegant way of doing this is to invoke the standard matrix inversion formula that results in

$$[I + 2\mu\, \underline{X}(j)\, \underline{X}^T(j)]^{-1} = I - \mathbf{c}\, \underline{X}(j)\, \underline{X}^T(j) ,$$

where $\mathbf{c} = 1 + \underline{X}^T(j)\, \underline{X}(j)$ (2.43a)

is a constant, and compute Eq. (2.42) according to

$$A(j) \triangleq I - c\,\underline{X}(j)\,\underline{X}^T(j) \tag{2.43b}$$

$$\underline{W}(j) = A(j)\,[\underline{W}(j-1) + 2\mu d(j)\,\underline{X}(j)] \tag{2.43c}$$

$$z(j) = d(j) - \underline{W}^T(j)\,\underline{X}(j)\ . \tag{2.43d}$$

But this still leaves considerable computation to be performed and consequently makes real-time signal processing difficult to achieve. A further simplification in computation is described later on.

In an implementation based on digital processors it is mainly the number of multiplications in terms of which the computational complexity is assessed. With a view to reducing the number of multiplications in the case of LMS algorithm, various alternative algorithms have been proposed in recent years. These mainly fall into two categories: what may be called, the "clipped-information" algorithms, and frequency domain algorithms. These are discussed next.

2.5 The Clipped-data LMS Algorithm

The clipped-information algorithms make use of infinitely quantized data or output in the weight update equation (2-12) thereby eliminating the need for multiplication. When the data is quantized or "clipped," the following algorithm, known as the clipped-data LMS (CLMS) algorithm [2-6], results:

$$y(j) = \sum_{i=0}^{n-1} w_i(j)\, x\,(j-i) \qquad (2.44a)$$

$$z(j) = d(j) - y(j) \qquad (2.44b)$$

$$w_i(j+1) = w_i(j) + 2\mu\, z(j)\, \mathrm{Sgn}\, x(j-i)\ . \qquad (2.44c)$$

This elimination of n multiplications in the above relations provides a distinct advantage in terms of speed of operation of the processor without any sacrifice in the level of convergence. However, the speed of adaptation is stated [2-6] to have been decreased in comparison with the LMS algorithm by a factor of $\pi/2$; it has also been mentioned that the steady-state values of the weights and the time constant depend on the amplitude levels of the reference signals rather than their power levels.

For the purpose of comparison with the other algorithms, the properties of the CLMS algorithm were derived. The derivation is included in Appendix A. It is seen that the derivation leads to a time constant different from that mentioned in [2-6]. The algorithm was simulated to confirm the results presented here. It was verified that for Gaussian inputs the actual time constant is given by the relation derived in the appendix.

Without loss of generality, the properties of the one-dimensional case of the LMS, MLMS and CLMS algorithms may be compared. These are given in Table 2.1.

Table 2.1 : Comparison of properties of algorithms.

Algorithm	Steady-state weight	Condition for convergence	Time constant
LMS	$R_x^{-1} R_{dx}$	$0 < \mu < \frac{1}{R_x}$	$\frac{1}{2\mu R_x}$
MLMS	$R_x^{-1} R_{dx}$	Always converges	$\frac{1}{2\mu R_x}$
CLMS	$R_x^{-1} R_{dx}$	$0 < \mu < \frac{\sigma_x}{R_x} \frac{\sqrt{\pi}}{\sqrt{2}}$	$\frac{\sigma_x}{2\mu R_x} \frac{\sqrt{\pi}}{\sqrt{2}}$

It is seen that all the three algorithms converge to the Wiener solution in the limit, under the stated assumptions. The time constants of the LMS algorithm and the MLMS algorithm are the same and that of the CLMS algorithm differs from the rest by a factor of $\sigma_x \frac{\sqrt{\pi}}{\sqrt{2}}$, whereas σ_x is the standard deviation of the reference input. In so far as the stability is concerned, the MLMS algorithm proves superior to the other two.

If, instead of the reference data, clipped output is used in the weight update equation (2.12), then, what is known as the hybrid (HYB) algorithm [2-6] results.

The computational saving in the CLMS algorithms has been achieved at the cost of speed of convergence, which may be disadvantageous in certain applications. The

frequency domain algorithms seek to overcome this difficulty for large values of n. These are described below.

2.6 Frequency Domain Algorithms

The frequency domain algorithms evaluate Eq. (2.12) using FFT techniques. Two algorithms have so far been proposed, one [2-7] using circular convolution and the other [2-8] using linear convolution. The difficulty with the former is that, in general, it does not ensure convergence of the weight vector to the Wiener solution; whereas the latter does ensure. The latter algorithm, known as the fast LMS (FLMS) algorithm, is described here.

Grouping the data in n-point blocks, the relation (2.12) may be iterated n times to obtain

$$y(mn+i) = \underline{W}^T(m)\, \underline{X}(mn+i) \tag{2.45}$$

$$z(mn+i) = d(mn+i) - y(mn+i) \tag{2.46}$$

$$\underline{W}(m+1) = \underline{W}(m) + 2\mu \sum_{i=0}^{n-1} z(mn+i)\, \underline{X}(mn+i) \tag{2.47}$$

where m is the block index, n the radix, and $0 \leq i \leq n-1$. It is assumed that the change in the weight vector over one data block is negligible so that the weights need be updated only at the end of each data block, leaving them fixed within a block.

The FLMS algorithm implements the above relations in the frequency domain. In as much as it is an exact implementation of the LMS algorithm in the frequency domain, its

properties are the same as the LMS algorithm implemented in the time domain. The FLMS algorithm [2-8] is summarized below.

The convolution in Eq. (2.45) and the correlation in Eq. (2.47) need be carried out using 2n-point FFT. Letting k denote the discrete frequency variable,

$$\underline{W}^T(k) \triangleq \text{FFT}\,[\underline{W}(m), \underline{0}]$$

$$\underline{X}^T(k) \triangleq \text{FFT}\,[x(mn-n), \ldots, x(mn), x(mn+1), \ldots, x(mn+n-1)]$$

$$\underline{Y}^T(m) \triangleq [y(mn), \ldots, y(mn+n-1)]$$

$$= \text{Last n terms of FFT}^{-1}\,[\underline{w}(k) \otimes \underline{X}(k)] \qquad (2.48a)$$

$$\underline{Z}^T(m) \triangleq [z(mn), \ldots, z(mn+n-1)]$$

$$= [d(mn) - y(mn), \ldots, d(mn+n-1) - y(mn+n-1)] \qquad (2.48b)$$

$$\underline{Z}(k) \triangleq \text{FFT}\,[\underline{0}, \underline{z}(m)]$$

$$\underline{W}(m+1) = \underline{W}(m) + 2\mu\,\underline{P}(m) \qquad (2.48c)$$

$\underline{P}(m) \triangleq$ First n terms of $\text{FFT}^{-1}\,[\underline{E}(k) \otimes \underline{X}(k)]$.

where $\otimes$ denotes the element-by-element multiplication of the two vectors.

2.7 An IIR LMS Algorithm

The adaptive filter is usually implemented by means of a programmable non-recursive (FIR) filter. Such a filter can produce only zeros with no poles in the filter transfer function. In applications where high selectivity is imperative, a relatively high-order transfer function has to be used to approximate the poles to a reasonable degree. For

the same filter specifications the required order in a non-recursive design can be as high as 5 to 10 times that in the recursive design [2-9]. Alternatively, a high-order FIR filter may equivalently be realized by a low-order IIR filter. This will result in considerable saving in computation as well as cost of implementation. Feintuch [2-10] has proposed an adaptive IIR LMS (ILMS) filter that will prove useful in certain noise cancelling applications. The ILMS algorithm is furnished by

$$y(j) = \underline{W}_1^T(j)\,\underline{X}(j) + \underline{W}_2^T(j)\,\underline{Y}(j) \tag{2.49a}$$

$$z(j) = d(j) - y(j) \tag{2.49b}$$

$$\underline{W}_1(j+1) = \underline{W}_1(j) + 2\mu_1\, z(j)\,\underline{X}(j) \tag{2.49c}$$

$$\underline{W}_2(j+1) = \underline{W}_2(j) + 2\mu_2\, z(j)\,\underline{Y}(j) \tag{2.49d}$$

where

$$\underline{X}^T(j) = [x(j), \ldots, x(j-n_1)]$$

$$\underline{Y}^T(j) = [y(j-1), \ldots, y(j-n_2)]$$

The block diagram of the resulting noise canceller is shown in Figure 2.4.

An examination of the algorithms discussed so far suggests new possibilities of computationally efficient algorithms. These are now outlined.

2.8 New Algorithms

Speed of a digital ANC is of concern when signals of large bandwidths have to be processed. For applications in which slow adaptation is acceptable, the CLMS algorithm has an edge over the others in terms of speed of processing. A

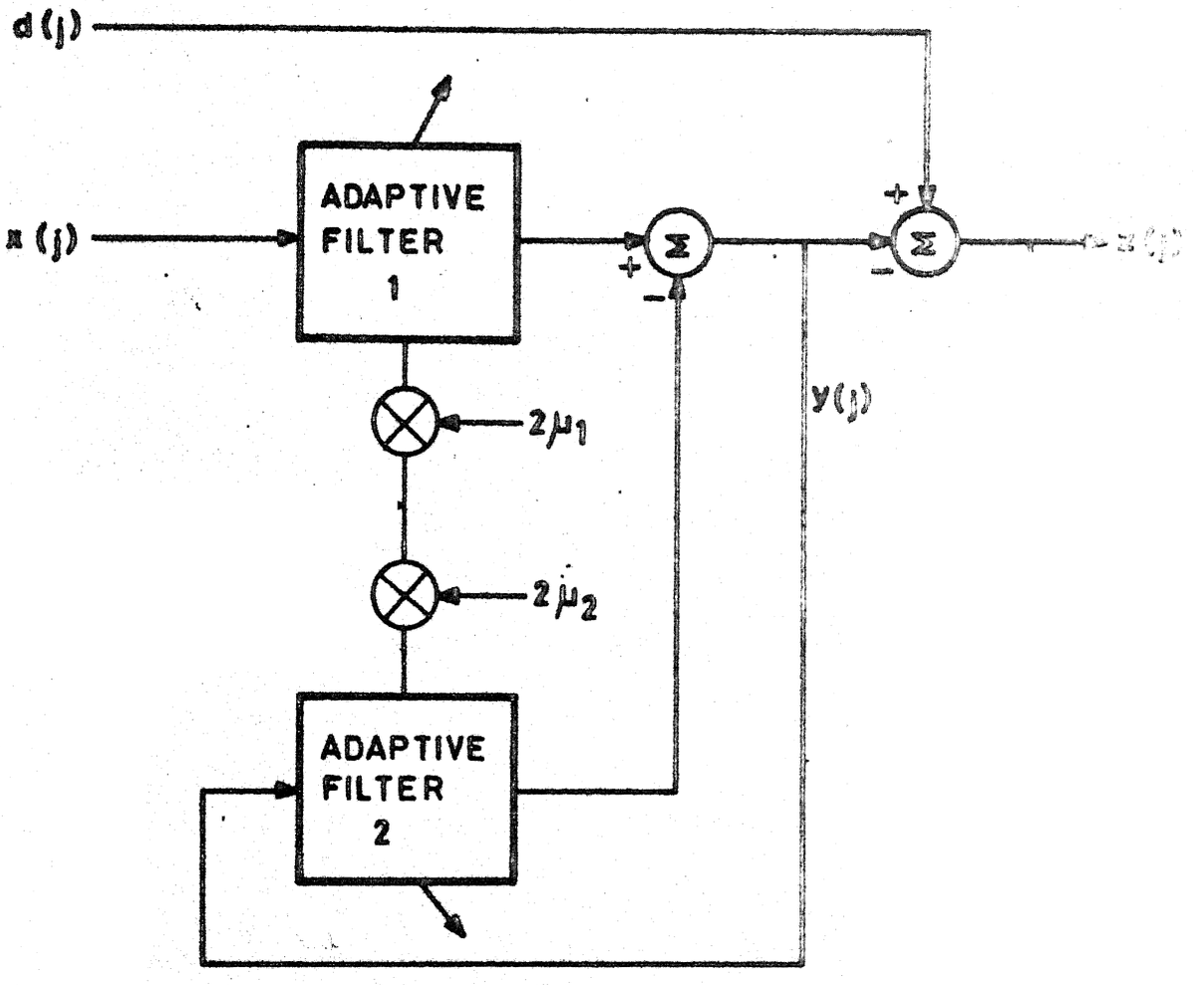


Fig. 2.4: Adaptive recursive LMS filter.

further improvement in speed can be obtained by implementing the FLMS algorithm partly in the sample domain and partly in the frequency domain. This strategy results in an algorithm that possesses significant computational advantage over all other FIR LMS algorithms, for large values of n.

Clipping the data in Eq. (2.47) gives

$$\underline{W}(m+1) = \underline{W}(m) + 2\mu \sum_{i=0}^{n-1} z(mn+i) \text{ Sgn } \underline{X}(mn+i) \qquad (2.50)$$

Equation (2.45) is evaluated in the frequency domain as given by Eq. (2.48a). After evaluating Eq. (2.46), the computation of Eq. (2.50) is carried out in the sample domain itself. Though it seems that n^2 sign checks would be required to compute Eq. (2.50), it can be implemented with only n sign checks. Therefore, evaluation of Eq. (2.45) by use of FFT algorithm, computing Eq. (2.46) and updating the weights in accordance with Eq. (2.50) is what constitutes a fast algorithm, herein called the fast clipped-data LMS (FCLMS) algorithm.

If, on the other hand, the output is clipped in the weight update equation, Eq. (2.47) becomes

$$\underline{W}(m+1) = \underline{W}(m) + 2\mu \sum_{i=0}^{n-1} [\text{Sgn } z(mn+i)] \underline{X}(mn+i) \, . \qquad (2.51)$$

This equation obviously requires n sign checks to update the weights. Equations (2.46), (2.48a) and (2.51) result in, what may be called, the fast HYB algorithm.

It is also possible to clip the data in Eqs. (2.49c) and (2.49d) so that considerable saving in computation results, though at the cost of speed of convergence. The resulting algorithm may be called the clipped-data IIR LMS (CILMS) algorithm. If, instead, the output is clipped in these relations, what may be called, the clipped-error IIR LMS algorithm results. If, in a similar way, the reference data are clipped in Eqs. (2.43a), (2.43b) and (2.43c), substantial saving in computation results. The consequent decrease in the speed of adaptation, in this case, may be compensated to some extent by increasing the value of μ as there is no constraint on it for convergence.

The algorithms proposed here have been listed in Appendix B. These were implemented on DEC-1090 system and tested for Gaussian inputs - white and coloured - and uniformly distributed inputs. They need be tested for other non-Gaussian inputs.

It is now examined how the algorithms considered so far compare with each other in regard to speed of computation. As pointed out earlier, the computational complexity of an LMS algorithm is primarily decided by the number of multiplications involved. Accordingly, this has been chosen as the basis of comparison. Also, a block of n iterations will be considered instead of one. Furthermore, it will be assumed that the computation time requirements for "house-keeping" operations in the FFT algorithm is negligible in relation to

that required for multiplications, and that each complex multiplication requires 4 real multiplications.

For real data, a 2n point FFT can be realized with an n point FFT and n complex multiplications [2-11]. Each n point FFT requires $\frac{n}{2} \log \frac{n}{2}$ complex multiplications. For the evaluation of Eq. (2.48a), for example, a total of $3(\frac{n}{2} \log \frac{n}{2} + n) + n$ complex multiplications will be required. Table 2.2 gives the relative computational requirements of various algorithms. The FIR LMS algorithms are compared with the LMS algorithm, whereas the CILMS algorithm is compared with the ILMS algorithm. It is seen that the CLMS and CILMS algorithms prove computationally efficient for small values of n, and the FCLMS and FCILMS algorithms for large values of n.

The algorithms discussed thus far in this section are by no means exhaustive. Over the years several algorithms have been proposed which, though, compute the Wiener solution in an iterative fashion, differ in some respect or the other. Noteworthy among them are the higher-order algorithms [2-12] which have weight vector convergence curves different from those of the LMS algorithm. However, these algorithms do not speed up convergence in an absolute sense. White [2-13] has reported a generalized IIR LMS algorithm of which Feintuch's algorithm discussed previously is a particular case. A comparative discussion about some of the earlier algorithms may be found in [2-3].

Table 2.2 : Comparison of Computation.

n	FLMS $\frac{(10 \log \frac{n}{2} + 28) \times n}{2n^2}$	CLMS $\frac{n^2}{2n^2}$	FCLMS $\frac{(6 \log \frac{n}{2} + 16) \times n}{2n^2}$	CILMS $\frac{2n^2}{4n^2}$
16	1.8125	0.5	1.0625	0.5
32	1.0625	"	0.6250	"
64	0.6094	"	0.3593	"
128	0.3438	"	0.2031	"
256	0.1914	"	0.1132	"
512	0.1055	"	0.0625	"
1024	0.0576	"	0.0341	"

SECTION 3

ADVANTAGES AND LIMITATIONS

As was pointed out in Section 1, the main advantage of ANC's lies in their ability to learn the signal characteristics and track them if they vary slower than the adaptation rate of the system. In circumstances where the noise cancelling technique is applicable, levels of noise rejection are often attainable that would be difficult or impossible to achieve by direct filtering. Examples include adaptive notch filter, self-tuning filter and bias weight filter. The advantages of ANC's are illustrated in Section 4.

Besides the requirement for a suitable reference input, the factors that limit the system performance are: uncorrelated noise components present in the inputs, signal components present in the reference input, finite length of the adaptive filter, misadjustment due to gradient estimation noise and errors arising from digital implementation.

The misadjustment was discussed in Section 2. The other performance limitations are described in this section. The discussion here closely follows [2-1]. For stationary stochastic inputs, the steady-state performance of adaptive filters closely approximates that of fixed Wiener filters, as was seen in Section 2. Thus the analysis of the shortcomings of ANC's is facilitated by the Wiener filter theory.

The Wiener filter is assumed to be unconstrained; i.e., it is composed of an infinitely long, two-sided (non-causal), tapped delay line. The optimal impulse response is obtained from the discrete Wiener-Hopf equation [see Eq. (2.8)]:

$$\sum_{i=-\infty}^{\infty} w_*(i)\, R_x(l - i) = R_{dx}(l) \tag{3.1}$$

where

$$R_x(l) \triangleq E[x(j)\, x(j + l)]$$

and

$$R_{dx}(l) \triangleq E[d(j)\, x(j + l)].$$

The z transform of Eq. (3.1) yields the transfer function:

$$w_*(z) = \frac{S_{dx}(z)}{S_x(z)} \tag{3.2}$$

where $S_x(z)$ and $S_{dx}(z)$, the power spectral densities, are the z transforms of $R_x(z)$ and $R_{dx}(z)$, respectively. The application of Wiener filter theory to noise cancelling is now considered.

3.1 Effect of Uncorrelated Noise Components in the Inputs

Figure 3.1 shows an adaptive noise canceller, which has, besides the usual inputs, a pair of noise components, $m_d(j)$ and $m_x(j)$, that are uncorrelated with

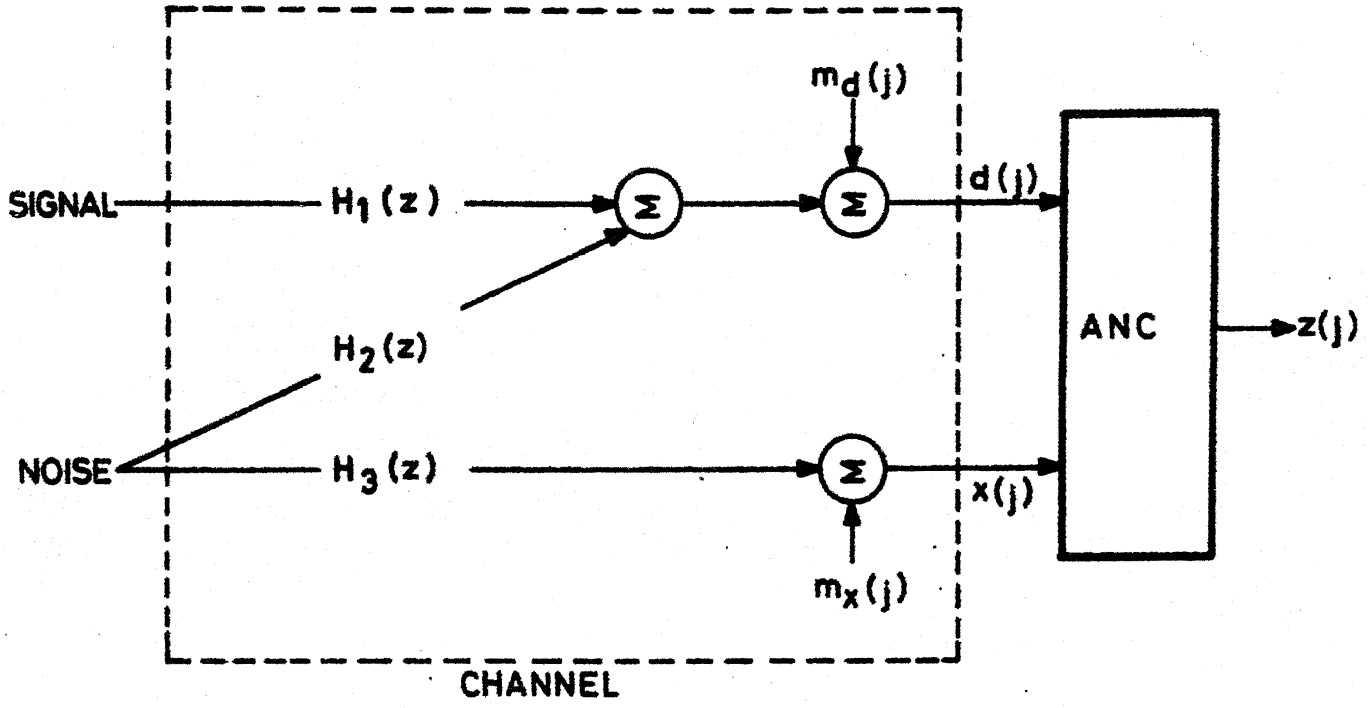


Fig. 3.1: ANC with uncorrelated noise components in the inputs.

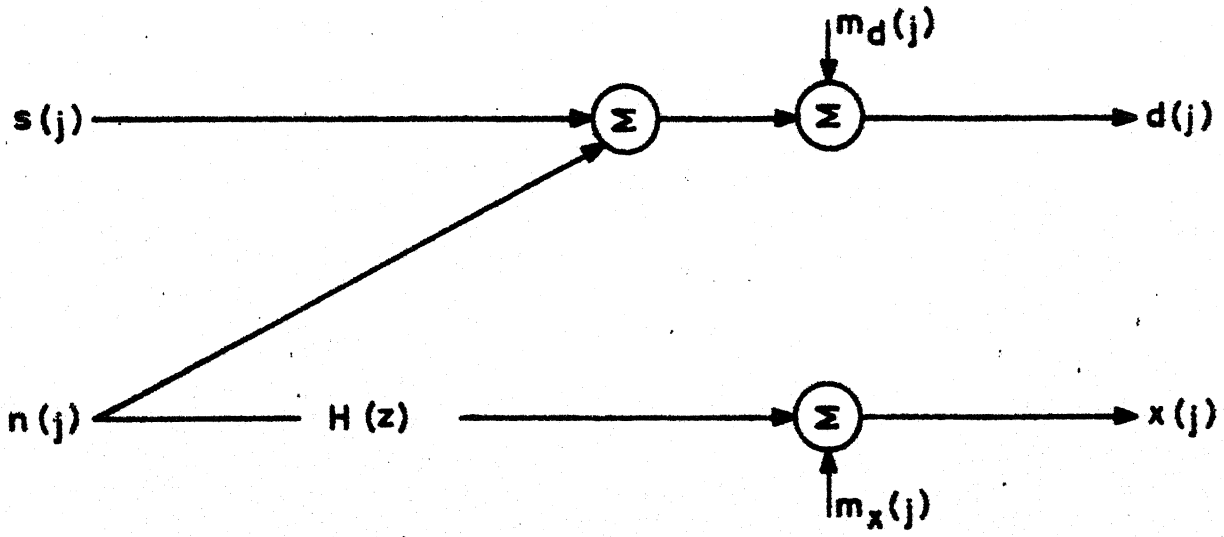


Fig. 3.2: Equivalent channel representation.

each other, with s, with n(j), and with h(j) * n(j). The uncorrelated noise components and the signal are assumed to have a finite power spectrum at all frequencies. For the purpose of analysis all noise propagation paths, represented by $H_2(z)$ and $H_3(z)$, are assumed to be equivalent to linear, time-invariant filters. The transfer function of the signal propagation path is represented by $H_1(z)$.

The pictorial model of the channel in Figure 3.1 may equivalently be represented by the model in Figure 3.2; where $H(z) = \frac{H_2(z)}{H_1(z)}$. The optimal, unconstrained transfer function of the adaptive filter, assuming that the adaptive process has converged and the minimum mse solution has been found, can be obtained from Eq. (3.2) as follows: The inputs to the ANC, from Figure 3.2, are

$$d(j) = s(j) + n(j) + m_d(j)$$

and

$$x(j) = h(j) * n(j) + m_x(j) .$$

Denoting the power spectral density of the noise $m_x(j)$ by $S_{m_x}(\bullet)$ and that of n(j) by $S_n(z)$, the filter's input spectrum, $S_x(z)$, can be written as

$$S_x(z) = S_{m_x}(z) + S_n(z) \, |H(z)|^2 . \qquad (3.3)$$

The cross-spectral density, $S_{dx}(z)$, is given by

$$S_{dx}(z) = S_n(z)\, H(z^{-1}) \,. \tag{3.4}$$

Combining Eqs. (3.2), (3.3) and (3.4), we obtain the Wiener transfer function

$$W_*(z) = \frac{S_n(z)\, H(z^{-1})}{S_{m_x}(z) + S_n(z)\, |H(z)|^2} \,. \tag{3.5}$$

Let the uncorrelated-to-correlated spectral density ratios of the inputs be defined as

$$A(z) = \frac{S_{m_d}(z)}{S_n(z)} \tag{3.6}$$

and

$$B(z) = \frac{S_{m_x}(z)}{S_n(z)\, |H(z)|^2} \,. \tag{3.7}$$

Combining Eqs. (3.5) and (3.7) yields

$$W_*(z) = \frac{1}{H(z)\,[1 + B(z)]} \,. \tag{3.8}$$

The performance of the noise canceller may be evaluated in terms of signal-to-noise density ratios, defined as

$$\rho_z(z) = \frac{\text{Output signal spectral density}}{\text{Output noise spectral density}}$$

and

$$\rho_d(z) = \frac{\text{Primary signal spectral density}}{\text{Primary noise spectral density}} \,.$$

Therefore,

$$\frac{\rho_z(z)}{\rho_d(z)} = \frac{S_n(z) + S_{m_d}(z)}{S_{n_z}(z)}$$

or

$$= \frac{S_n(z)\,[1 + A(z)]}{S_{n_z}(z)} \qquad (3.9)$$

where $S_{n_z}(z)$ is the output noise spectral density corresponding to the output noise, $n_z(j)$, given by

$$n_z(j) = m_d(j) - w_*(j) * m_x(j) + [1 - h(j) * w_*(j)] * n(j) .$$

The power spectral density of the output noise is

$$S_{n_z}(z) = S_{m_d}(z) + |w_*(z)|^2 S_{m_x}(z) + |[1 - H(z)\, w_*(z)]|^2 S_n(z) . \qquad (3.10)$$

Combining Eqs. (3.6), (3.8) and (3.10) yields

$$S_{n_z}(z) = S_n(z)\, A(z) + S_n(z) \frac{B(z)}{1 + B(z)} . \qquad (3.11)$$

From Eqs. (3.9) and (3.11), we obtain

$$\frac{\rho_z(z)}{\rho_d(z)} = \frac{[1 + A(z)]\,[1 + B(z)]}{A(z) + A(z)\,B(z) + B(z)} .$$

The above expression provides a general description of ideal noise canceller performance, and as such

enables one to estimate the level of noise reduction possible in an ideal noise cancelling system with inputs containing uncorrelated noise components. It is apparent from the equation that the level of noise reduction is limited by the uncorrelated-to-correlated noise density ratios at the primary and reference inputs. This is perhaps best illustrated by the following specific cases:

Case 1 When $A(z) = 0$,

$$\frac{\rho_z(z)}{\rho_d(z)} = 1 + \frac{1}{B(z)} .$$

Case 2 When $B(z) = 0$,

$$\frac{\rho_z(z)}{\rho_d(z)} = 1 + \frac{1}{A(z)} .$$

Case 3 When $A(z) = B(z) = 0$,

$$\frac{\rho_z(z)}{\rho_d(z)} = \infty .$$

These cases demonstrate the desirability of low levels of uncorrelated noise components in the inputs. The smaller the values of $A(z)$ and $B(z)$, the larger is the ratio $\rho_z(z)/\rho_d(z)$ and, hence, the more effective is the action of noise canceller. It may be noted that when $m_x(j) = 0$, Eq. (3.5) reduces to

$$W_*(z) = \frac{1}{H(z)} ,$$

which shows that the adaptive filter essentially adapts to the inverse of the channel characteristics.

3.2 Effect of Signal Components in the Reference Input

In certain situations the available reference input to an ANC may contain low-level signal components, besides the usual correlated and uncorrelated noise components. These signal components will, no doubt, cause some cancellation of the primary input, thereby distorting it. A quantitative analysis of such a distortion is considered here. For simplicity, it is assumed that the uncorrelated noise components in the inputs are absent. It is also assumed that the signal propagation path between the source and the reference input (see Figure 3.2) is characterized by a linear, time-invariant filter, $G(z)$.

The reference power spectral density, $S_x(z)$, in this case, is given by

$$S_x(z) = |G(z)|^2 S_s(z) + |H(z)|^2 S_n(z)$$

and the cross-spectral density, S_{dx}, by

$$S_{dx}(z) = G(z^{-1}) S_s(z) + H(z^{-1}) S_n(z).$$

The Wiener transfer function is, therefore,

$$W_*(z) = \frac{G(z^{-1}) S_s(z) + H(z^{-1}) S_n(z)}{|G(z)|^2 S_s(z) + |H(z)|^2 S_n(z)} . \qquad (3.12)$$

The spectral density of the signal component in the output is

$$|1 - G(z)\, W_*(z)|\, S_s(z)$$

and that of the noise component in the output

$$|1 - H(z)\, W_*(z)|\, S_n(z) .$$

The output signal-to-noise density ratio is, therefore,

$$\rho_z(z) = \frac{|1 - G(z)\, W_*(z)|\, S_s(z)}{|1 - H(z)\, W_*(z)|\, S_n(z)} .$$

Substituting for $W_*(z)$ and cancelling the common terms yield

$$\rho_z(z) = \frac{|H(z)|^2\, S_n(z)}{|G(z)|^2\, S_s(z)} . \qquad (3.13)$$

The spectral density of the signal component in the reference input is

$$|G(z)|^2\, S_s(z)$$

and that of the noise component in the reference input

$$|H(z)|^2\, S_n(z) .$$

The signal-to-noise density ratio, $\rho_x(z)$, at the reference input is, therefore,

$$\rho_x(z) = \frac{|G(z)|^2\, S_s(z)}{|H(z)|^2\, S_n(z)} . \qquad (3.14)$$

From Eqs. (3.13) and (3.14), we obtain

$$\rho_z(z) = \frac{1}{\rho_x(z)} . \tag{3.15}$$

The above relation shows that, assuming the Wiener solution vector to be unconstrained and the noises in the two inputs to be mutually correlated, the signal-to-noise density ratio at the noise canceller output is the reciprocal of the signal-to-noise density ratio at the reference input. If $\rho_x(z) = 0$, then $\rho_z(z) = \infty$. Thus, the lower the signal components in the reference input, the better is the performance of the noise canceller.

The output signal distortion caused by the signal component propagated through the adaptive filter is measured in terms of a dimensionless quantity, D(z), which is defined as the ratio of the spectral density of the output signal component propagated through the adaptive filter to the spectral density of the signal component at the primary input; i.e.

$$D(z) = |G(z)\, W_*(z)|^2 . \tag{3.16}$$

If $|G(z)|$ is small, it may be reasonable to assume that $|G(z)|^2 \underline{\sim} 0$. Then, from Eq. (3.12),

$$D(z) \underline{\sim} \left|\frac{G(z)}{H(z)}\right|^2$$

or, alternatively,

$$D(z) \simeq \frac{\rho_x(z)}{\rho_d(z)} .$$

The above relation means that, under the stated assumptions, low signal distortion results from a high signal-to-noise density ratio at the primary input and a low signal-to-noise density ratio at the reference input. This shows that signal components of low signal-to-noise ratio at the reference input do not render the application of adaptive noise cancelling useless. If the reference input contained only signal components but no noise components, correlated or uncorrelated, then the signal components would completely be cancelled. However, if the reference input is properly derived, this condition will not occur.

3.3 Finite Length, Causal Approximation

The Wiener solution considered so far in this section is based on unconstrained Wiener filter theory; i.e., the filter calls for an infinitely long, two-sided tapped delay line, which is, of course, not feasible to realize. Since impulse responses of practical systems tend exponentially to zero with time, approximate realizations by means of finite length tapped-delay lines do become feasible.

In circumstances where a noncausal response is desired, a suitable delay may be provided in the primary

input so as to cause the peak of the impulse response to be located at the centre of the delay line. It has been pointed out [2-1] that a delay equal to about half the time delay of the adaptive filter produces the minimum output noise power.

3.4 Errors Arising from Digital Implementation

In an implementation based on digital hardware, errors due to truncation or roundoff are inevitable. When analog signals are processed, the quantization becomes yet another source of error. If these errors are not kept within bounds, the adaptive system might become unstable. It has been shown [3-1] that if the adaptation constant and the upper bounds on errors arising from quantization and roundoff are sufficiently small, then the convergence of the weight vector is ensured. An exact expression for the condition may be found in [3-1].

To summarize, the analysis presented in this section emphasizes the requirement for a suitable reference input. In general, some ingenuity is required in deriving the reference input such that the undesirable components are kept at a low level. Also, it is possible to realize a non-causal response by providing an appropriate delay in the primary input. Finally, the parameter μ has to be so chosen as to, besides ensuring convergence, keep the errors arising from digital implementation and misadjustment sufficiently low on the one hand and the adaptive process sufficiently fast on the other.

SECTION 4

APPLICATIONS AND SIMULATION RESULTS

Adaptive noise cancelling has been used in speech signal processing, pattern recognition, adaptive antennas, and medical electronics; the technique can also be used in the removal of power line interferences, tape hum, and in the detection of very-low-level signals corrupted by broad-band noise [2-1, 4-1]. In this section, several typical applications of ANC's are discussed and simulation results presented, which show the wide range of applicability of noise cancellers. The LMS algorithm was used in the simulations, which were carried out on the DEC-1090 system, using Fortran-10 language. For simplicity, the uncorrelated noise components in the inputs and the signal components in the reference input are assumed absent, though their presence would alter the results presented here only to the extent discussed in Section 3.

4.1 The ANC as a Notch Filter

The conventional method of eliminating a single frequency sinusoidal interference from a signal is through the use of a notch filter. The ANC could also be employed to remove an unwanted component from an incoming signal. The advantages of such an adaptive notch filter are: easy

control of bandwidth, an infinite null, and the ability to track the exact frequency of interference.

The primary signal can be of any kind - deterministic or stochastic. The noise in the primary input is assumed a $\cos(\omega_o t + \theta)$; here analog variables are used for notational convenience. The reference component is denoted by $b \cos(\omega_o t + \phi)$. Two degrees of freedom or taps are required for the filter to achieve the necessary gain and phase for cancellation. Since deterministic signals are involved here, the problem of noise cancelling may be reformulated as one of minimizing the sum of squared-errors over time. Assuming that the primary input is zero-mean ergodic, the optimum weight vector may be found by minimizing the sum of squared errors,

$$\lim_{m \to \infty} \frac{1}{m} \sum_{j=0}^{m-1} e^2(j) = \lim_{m \to \infty} \frac{1}{m} \sum_{j=0}^{m-1} [d(j) - \underline{W}^T \underline{X}(j)]^2 .$$

Here the signals have been assumed one-sided.

Following the same procedure as in Section 2, the optimum weight vector may be obtained by solving the linear simultaneous equations

$$R_x \underline{W}_* = \underline{R}_{dx} \tag{4.1}$$

where

$$R_x \triangleq \lim_{m \to \infty} \frac{1}{m} \sum_{j=0}^{m-1} \underline{X}(j) \underline{X}^T(j) \tag{4.2}$$

and

$$\underline{R}_{dx} \triangleq \lim_{m \to \infty} \frac{1}{m} \sum_{j=0}^{m-1} d(j)\, \underline{X}(j) \qquad (4.3)$$

where the relations (4.2) and (4.3) are similar to the empirical relations that result from the assumption of ergodicity of the stochastic inputs.

If it is assumed that the signal is uncorrelated with the noise, then Eq. (4.3) may be rewritten as

$$\underline{R}_{dx} = \lim_{m \to \infty} \frac{1}{m} \sum_{j=0}^{m-1} n_d(j)\, \underline{X}(j) \; .$$

It may be noted that $n_d(j)$ and $n_x(j)$, where $n_x(j) = x(j)$, are deterministic and Eqs. (4.2) and (4.3) define the corresponding discrete-time correlation functions. For periodic signals with period m, these equations may be rewritten as

$$R_x = \frac{1}{m} \sum_{j=0}^{m-1} \underline{X}(j)\, \underline{X}^T(j)$$

and

$$\underline{R}_{dx} = \frac{1}{m} \sum_{j=0}^{m-1} n_d(j)\, \underline{X}(j) \; .$$

Since two taps will suffice in the case of single frequency sinusoid, the correlation functions reduce to

$$R_x = \begin{bmatrix} \frac{b^2}{2} & \frac{b^2}{2} \cos \omega_o T \\ \frac{b^2}{2} \cos \omega_o T & \frac{b^2}{2} \end{bmatrix}$$

and

$$\underline{R}_{dx} = \begin{bmatrix} \frac{ab}{2}\cos(\theta - \phi) \\ \frac{ab}{2}\cos(\omega_o T - \theta + \phi) \end{bmatrix}$$

where T is the delay between the taps. If the delay is chosen as T'/4, where T' is the period of the signal, then the optimum weights are furnished by

$$\underline{W} = \begin{bmatrix} \frac{a}{b}\cos(\theta - \phi) \\ \frac{a}{b}\sin(\theta - \phi) \end{bmatrix}. \qquad (4.4)$$

To illustrate the effectiveness of the single-frequency noise canceller, a simulation was carried out with

$$d(j) = s(j) + 1.5 \sin\left(\frac{\omega_o T'}{9} j\right)$$

$$x(j) = \sin\left(\frac{\omega_o T'}{9} j + \frac{2\pi}{9}\right)$$

$\mu = 0.0005$, and $n = 2$, where $s(j)$ is a coloured Gaussian sequence obtained by passing a white Gaussian sequence (0, 1) [i.e., with zero mean and unit variance] through a single-pole low pass filter with the pole at 0.5. The values of the theoretical as well as the simulated weights are tabulated in page 4-18 [Table 4.1]. The simulated weights are in close agreement with the theoretical ones. The plots of the signals taken at various nodes after the steady-state had been reached are shown in Figures 4.1-1 to

4.1-6. The numbers to the left of the plots represent the amplitudes of the corresponding sequences. Also the number of '+' signs against each number is proportional to the corresponding amplitude.

Approximate expressions for the notch bandwidth and quality factor are given by [2-1]

$$\text{Notch bandwidth} \simeq \frac{2\mu b^2}{T}$$

and

$$\text{Quality factor} \simeq \frac{\omega_o T}{2\mu b^2},$$

respectively. From these expressions, it is seen that by an appropriate choice of μ, a narrow and infinite null can be achieved in the spectrum of the primary input. It may be noted that in case the delay is not chosen as $T'/4$, the weights will converge to values different from those given by Eq. (4.4).

In a second simulation, 512 discrete sinusoids were added to form the primary input, and the reference input was one of the primary components, but with a different phase and amplitude. The normalized power spectrum at the output after the weights had converged is shown in Figure 4.1-7. The plot shows an excellent rejection of the required signal component. When the reference signal frequency was slightly varied, even then there was cancellation, but the weights converged to a dynamic rather than a static solution

"tumbling" at the difference frequency; in other words, the ANC behaves like a modulator converting the reference frequency into primary frequency.

In a third simulation, an uncorrelated Gaussian noise sequence was used as the primary input. The reference input and the processing parameters were the same as in the first simulation. An ensemble average of 200 power spectra at the noise canceller output is shown in Figure 4.1-8. The resolution of the FFT used to obtain the power spectra was 512, what are known as in the jargon, spectral bins; i.e., the FFT was of 512 points. It was observed during the simulation that a fast adaptation leads to cancellation of adjacent frequency components. However, if it is kept slow, the effect is generally insignificant.

The results presented so far demonstrate the effectiveness of single frequency noise cancellers. In a similar way, a multiple frequency noise canceller can also be constructed. The second simulation was modified to remove three more components from the primary input. Figure 4.1-9 shows the corresponding power spectrum.

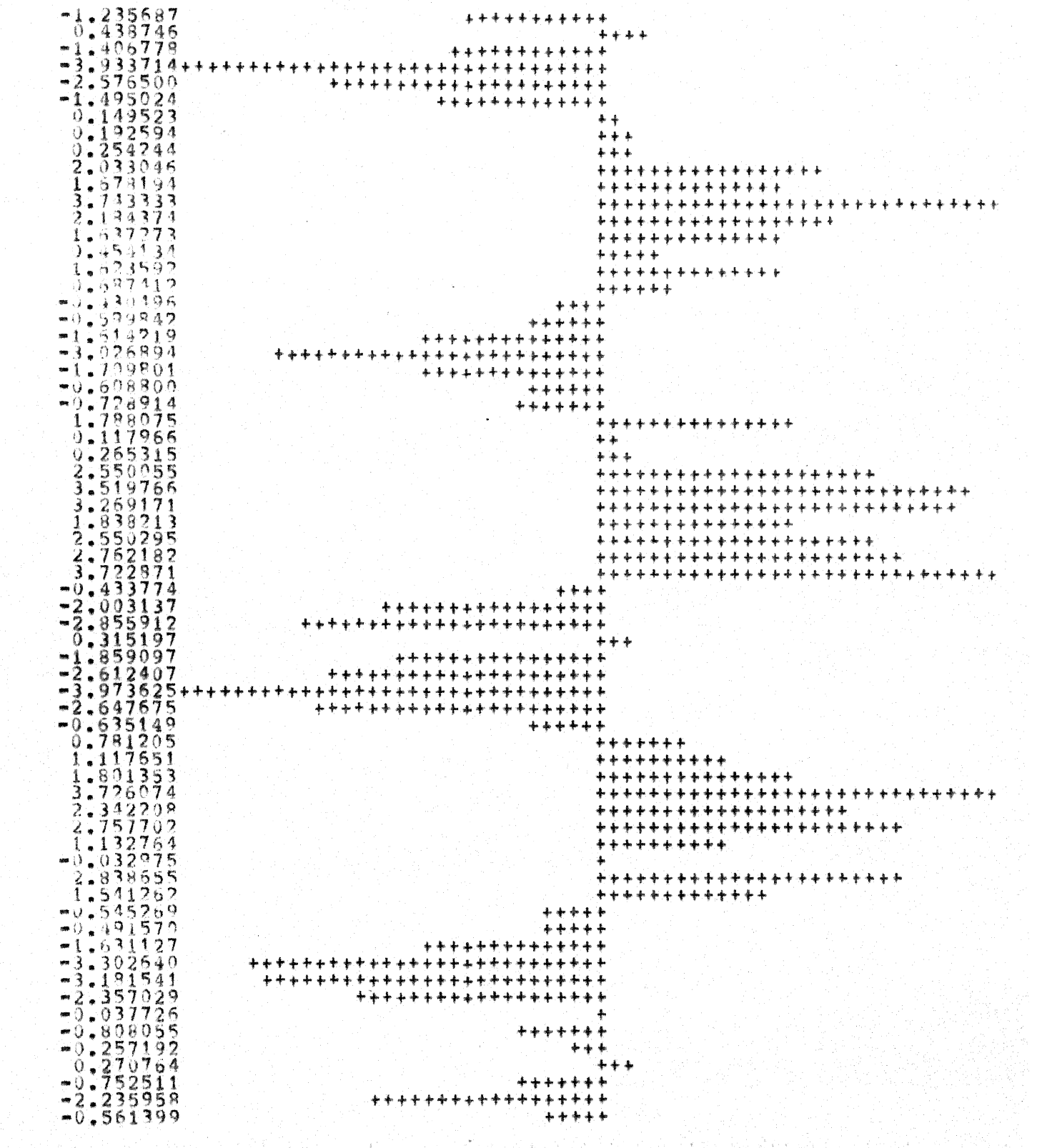


Fig. 4.1-1 : Primary input d(i).

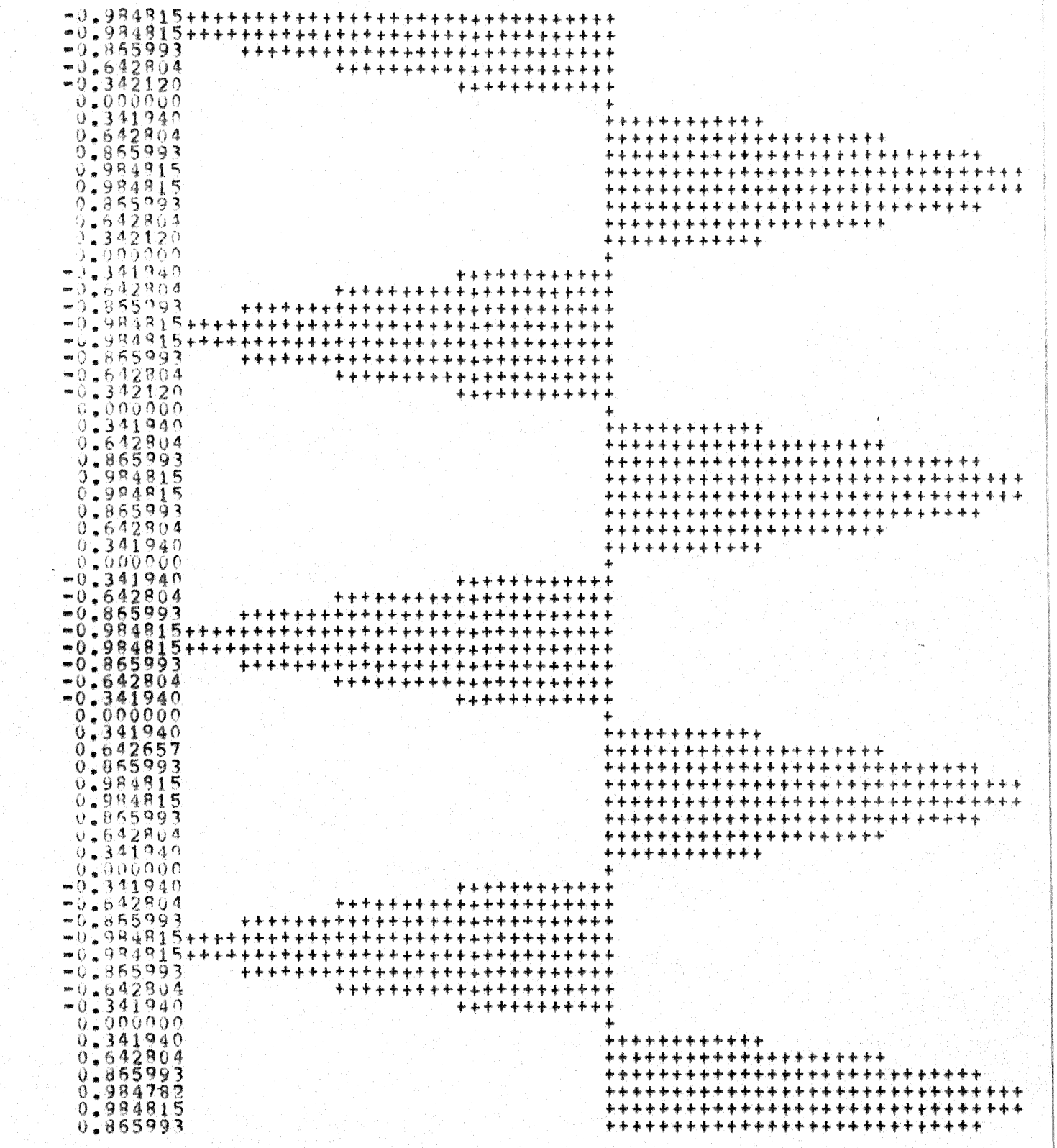


Fig. 4.1-2 : Reference input x(1).

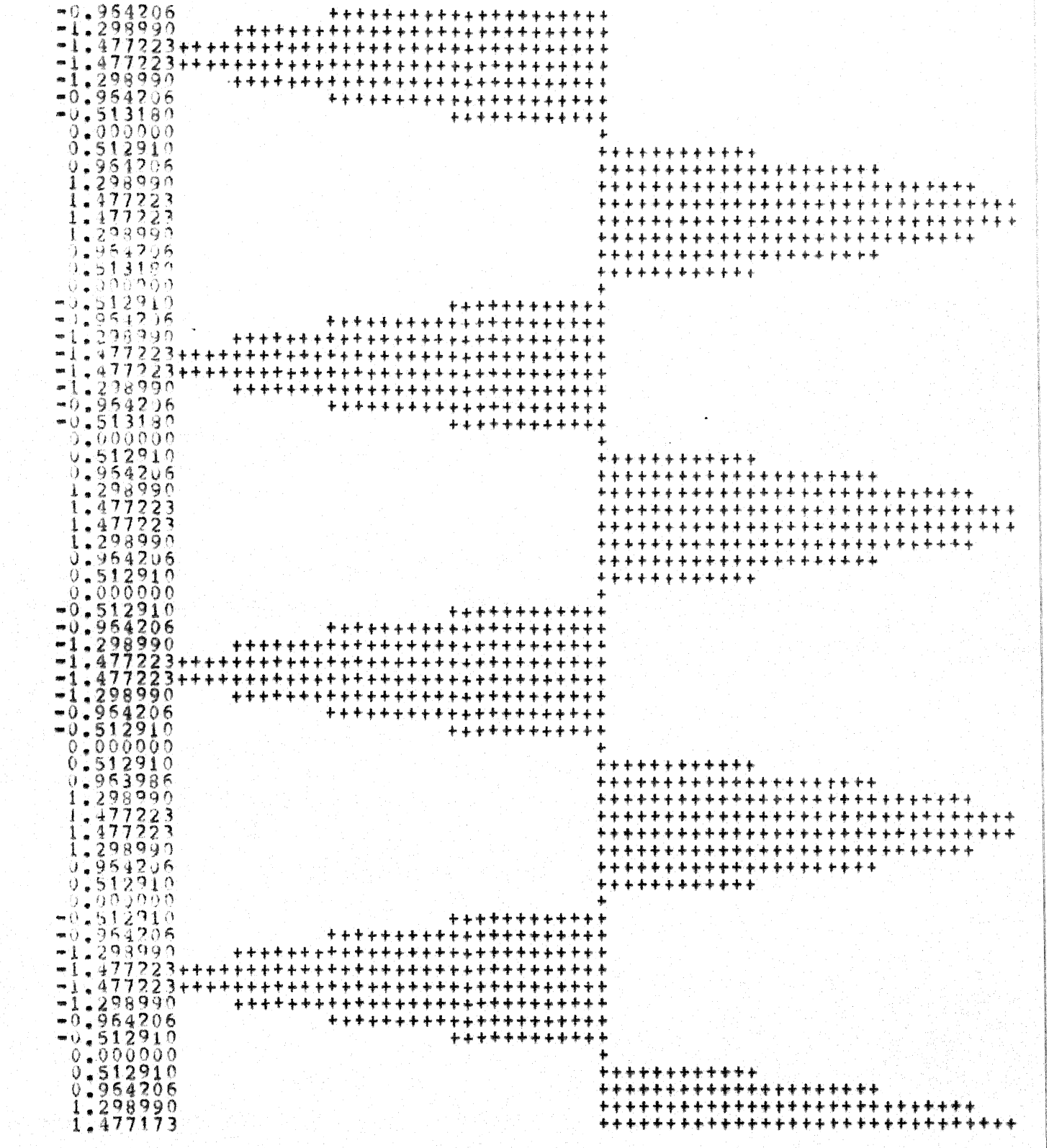


Fig. 4.1-3 : Primary noise $n_d(j)$.

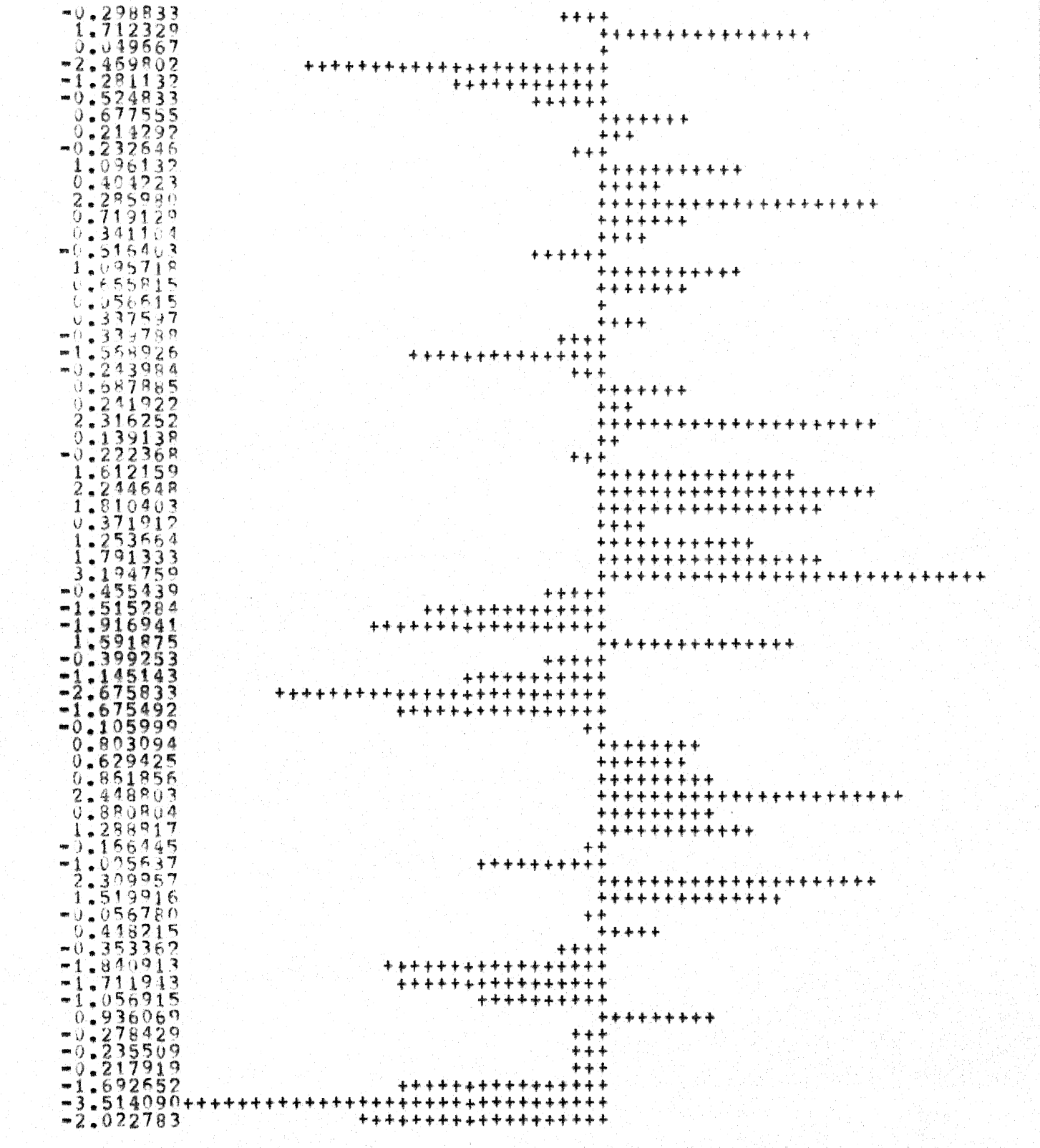


Fig. 4.1-5 : ANC output z(j).

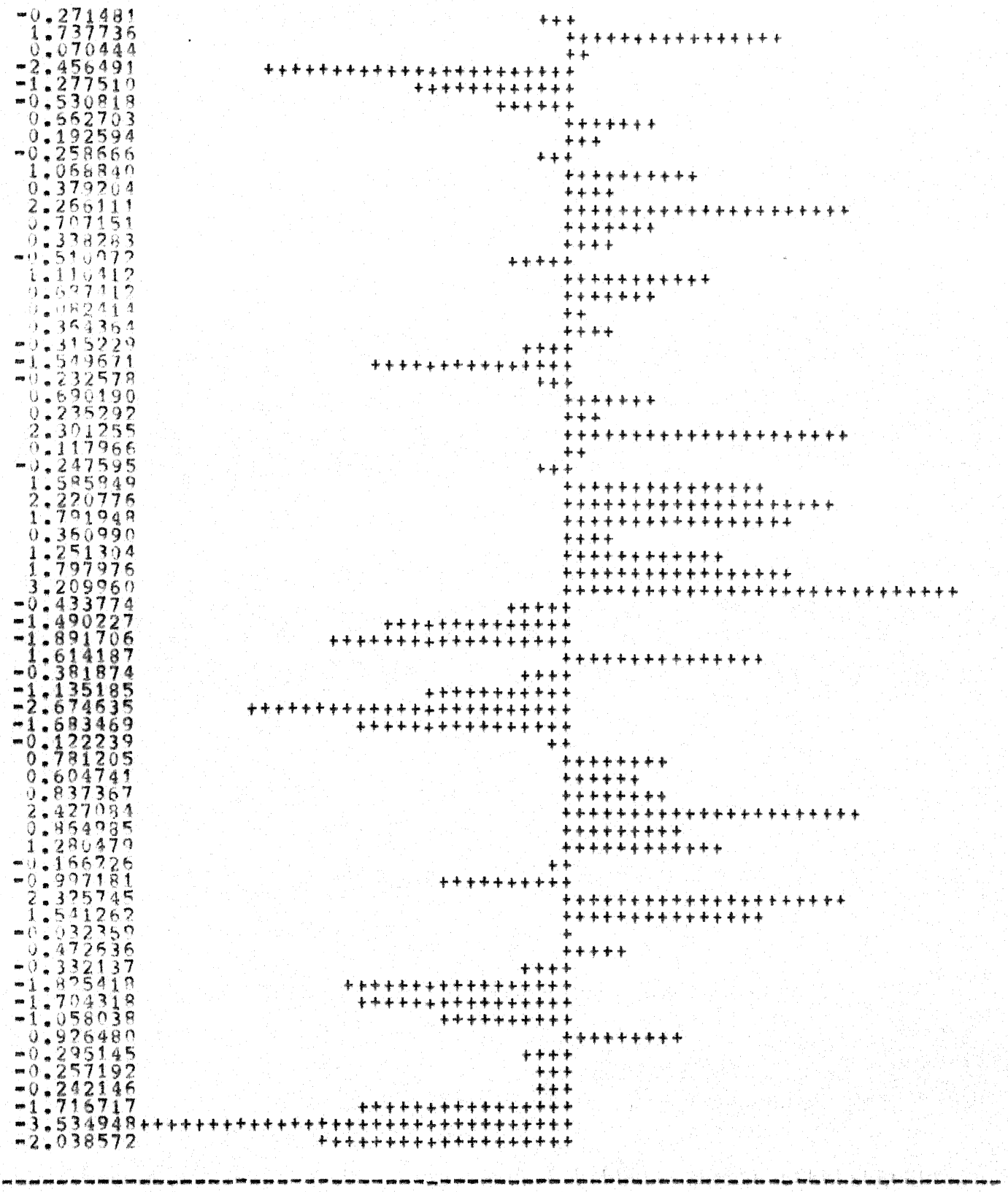


Fig. 4.1-6 : Signal s(1).

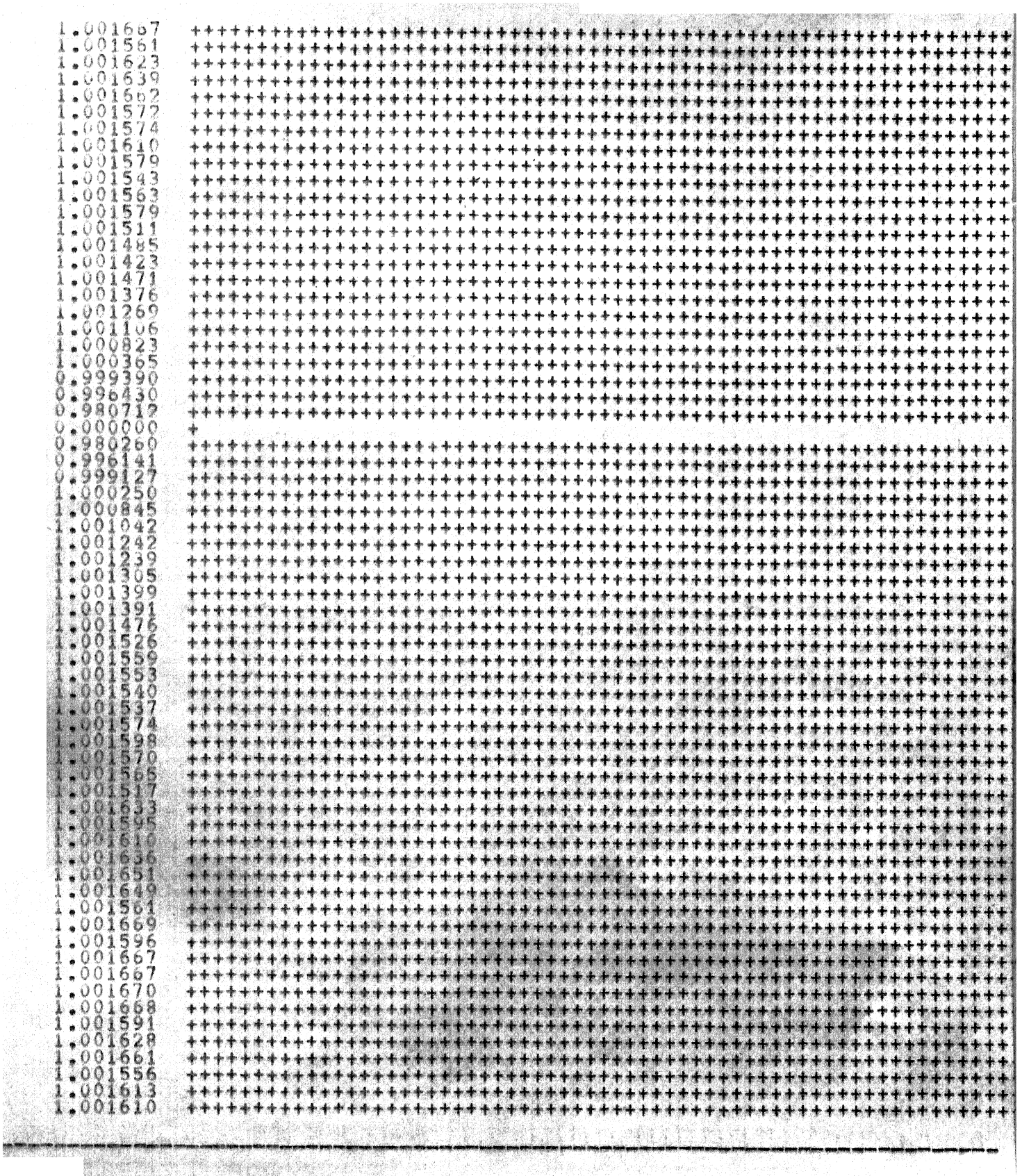


Fig. 4.1-7 : Normalized power spectrum - single notch.

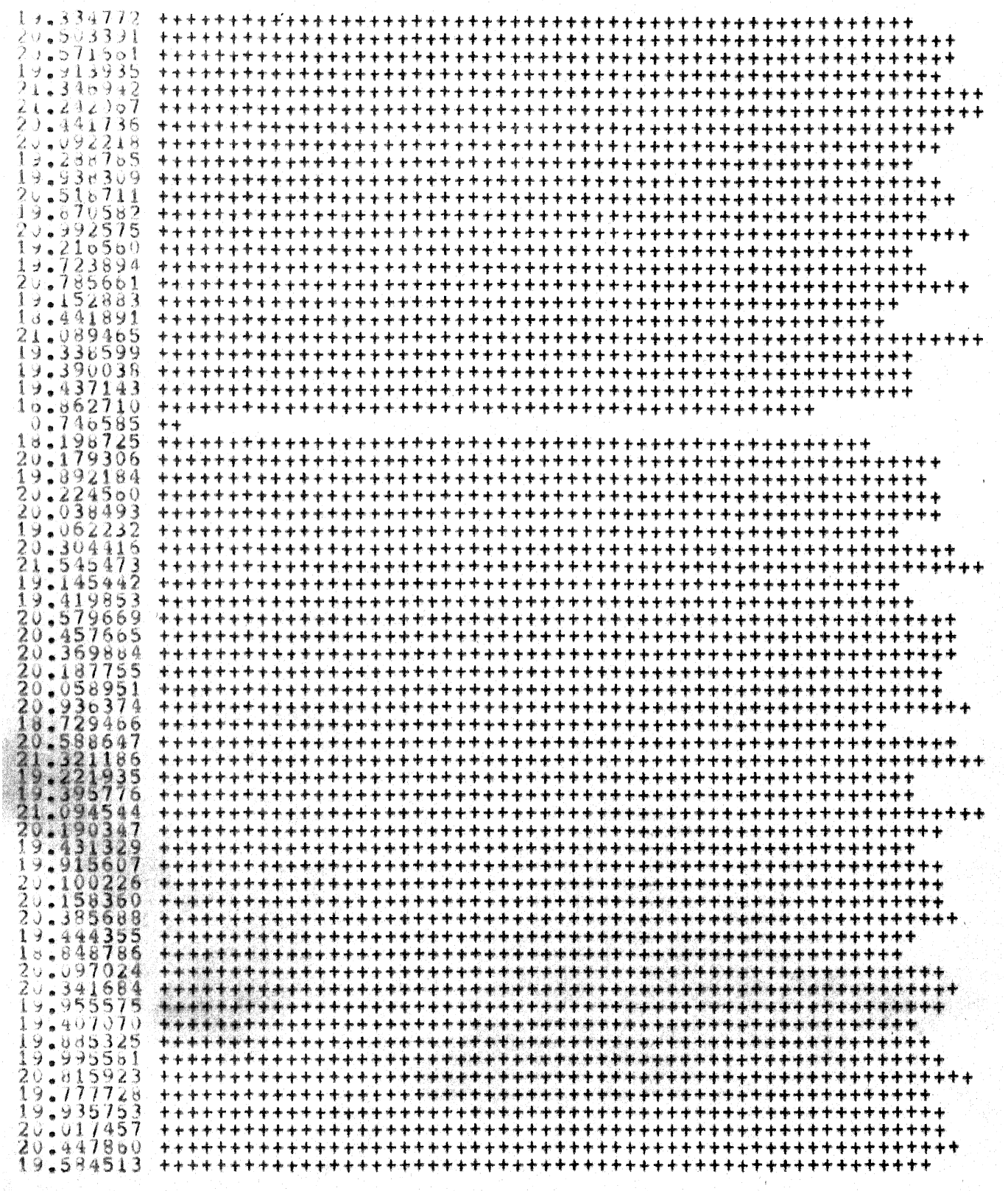


Figure 4.1-8 : Notch in White Gaussian Noise.

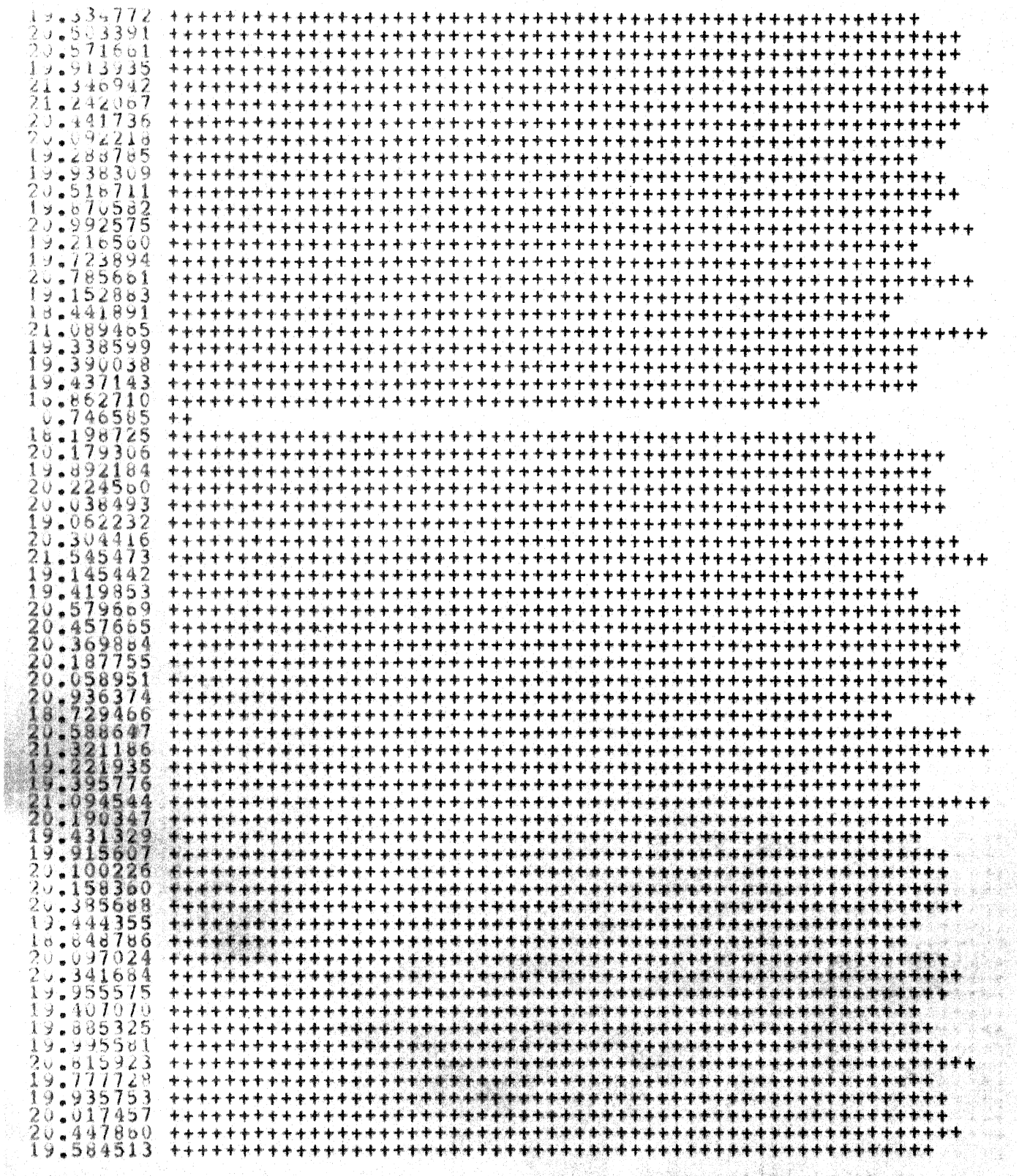


Figure 4.1-8 : Notch in White Gaussian Noise.

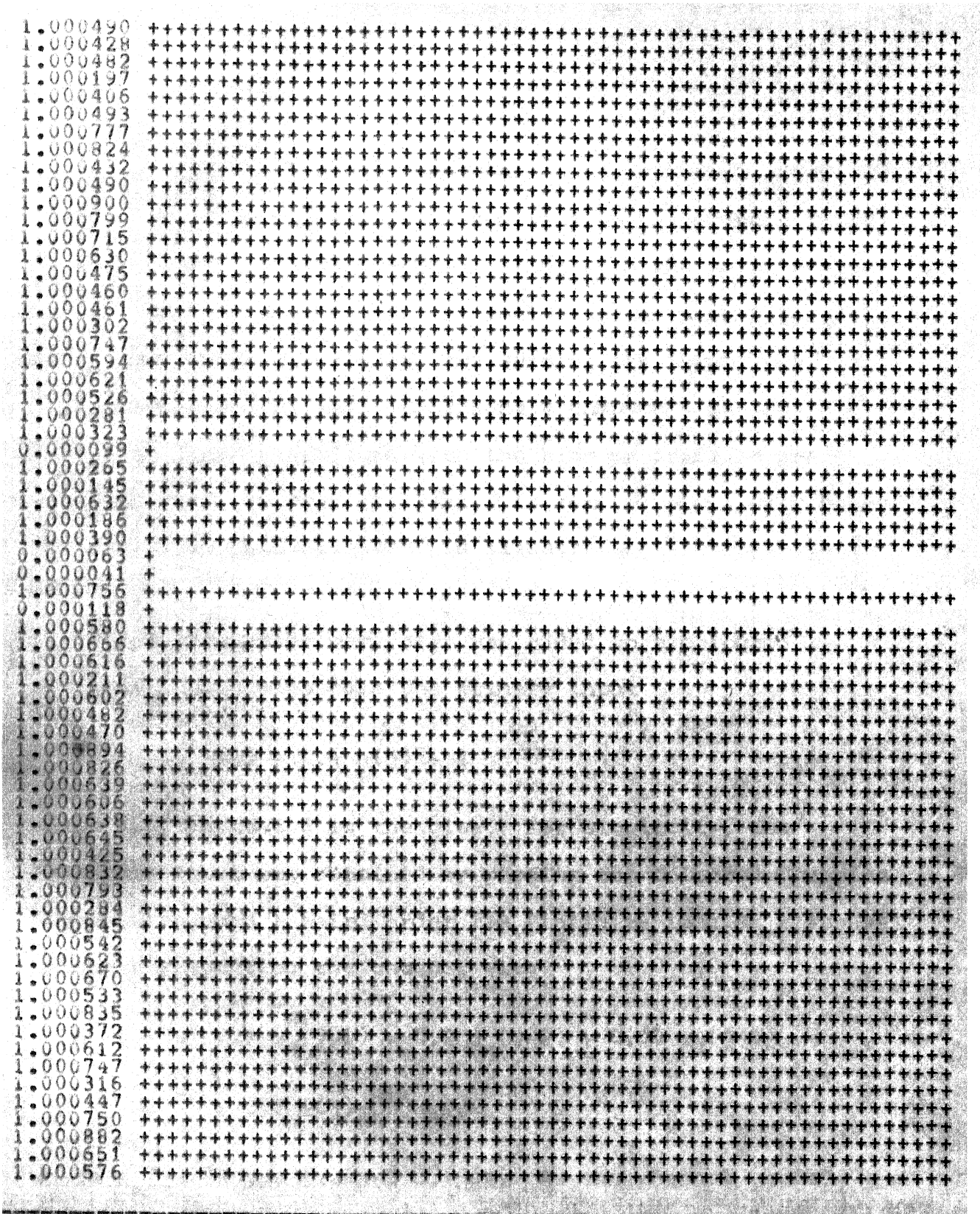


Fig. 4.1-9 : Normalised power spectrum - multiple notch.

4.2 The ANC as a Highpass Filter

It is also possible to remove from the primary input any slow drift or bias component along with the noise components. This is accomplished by including one more filter weight, known as bias weight, and setting its input to a constant value that will represent a component in the reference input correlated with the bias or drift in the primary input, so that the latter is removed along with other correlated components. This is illustrated in the present example.

A triangular wave s(j) was added to a coloured Gaussian sequence to form the primary input

$$d(j) = s(j) + 1.4\, n(j) + 2$$

where the constant 2 is the bias. For simplicity, the reference input was obtained by passing the coloured Gaussian sequence through a third order IIR (infinite impulse response) filter representing the channel with

$$n_x(j) = 0.7\, n_x(j-1) + 0.2\, n_x(j-2) + 0.05\, n_x(j-3) + 0.7\, n(j)$$

where $n_x(j)$ is the channel output. As the channel characteristics are known, it can be shown that the theoretical optimum weight vector given by

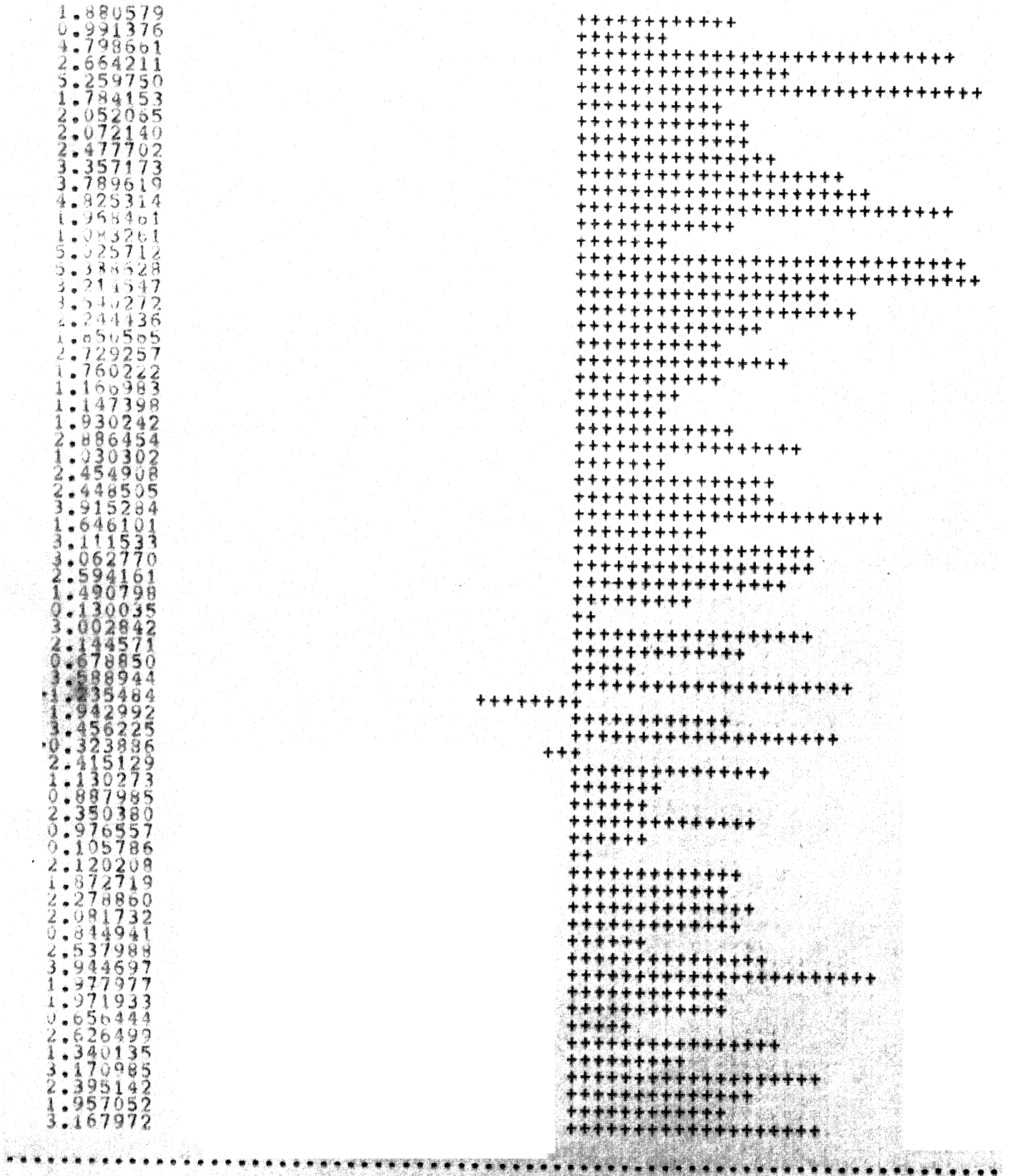


Fig. 4.2-1 : Primary input a(j).

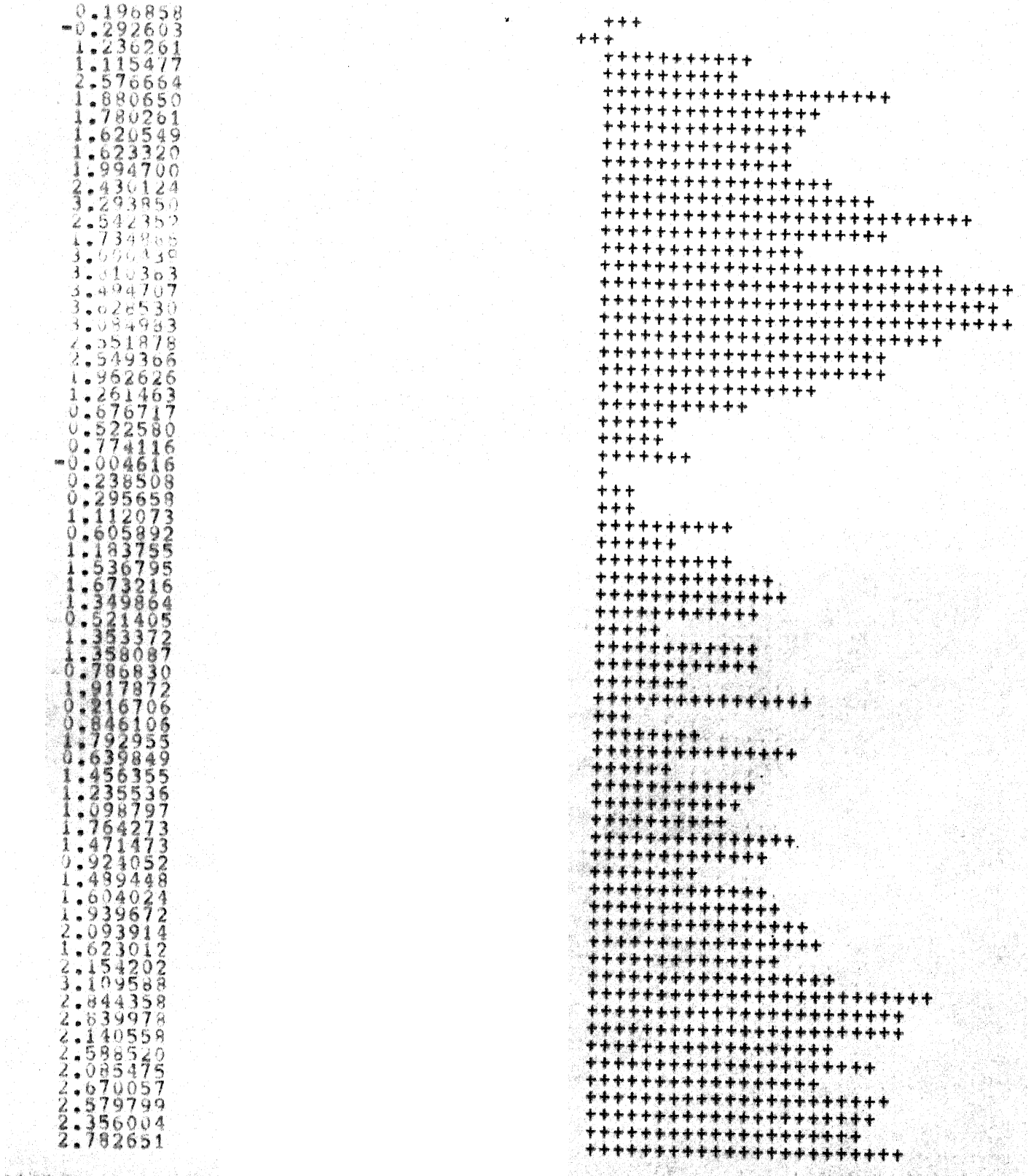


Fig. 4.2-2 : ReferenCe input x(j).

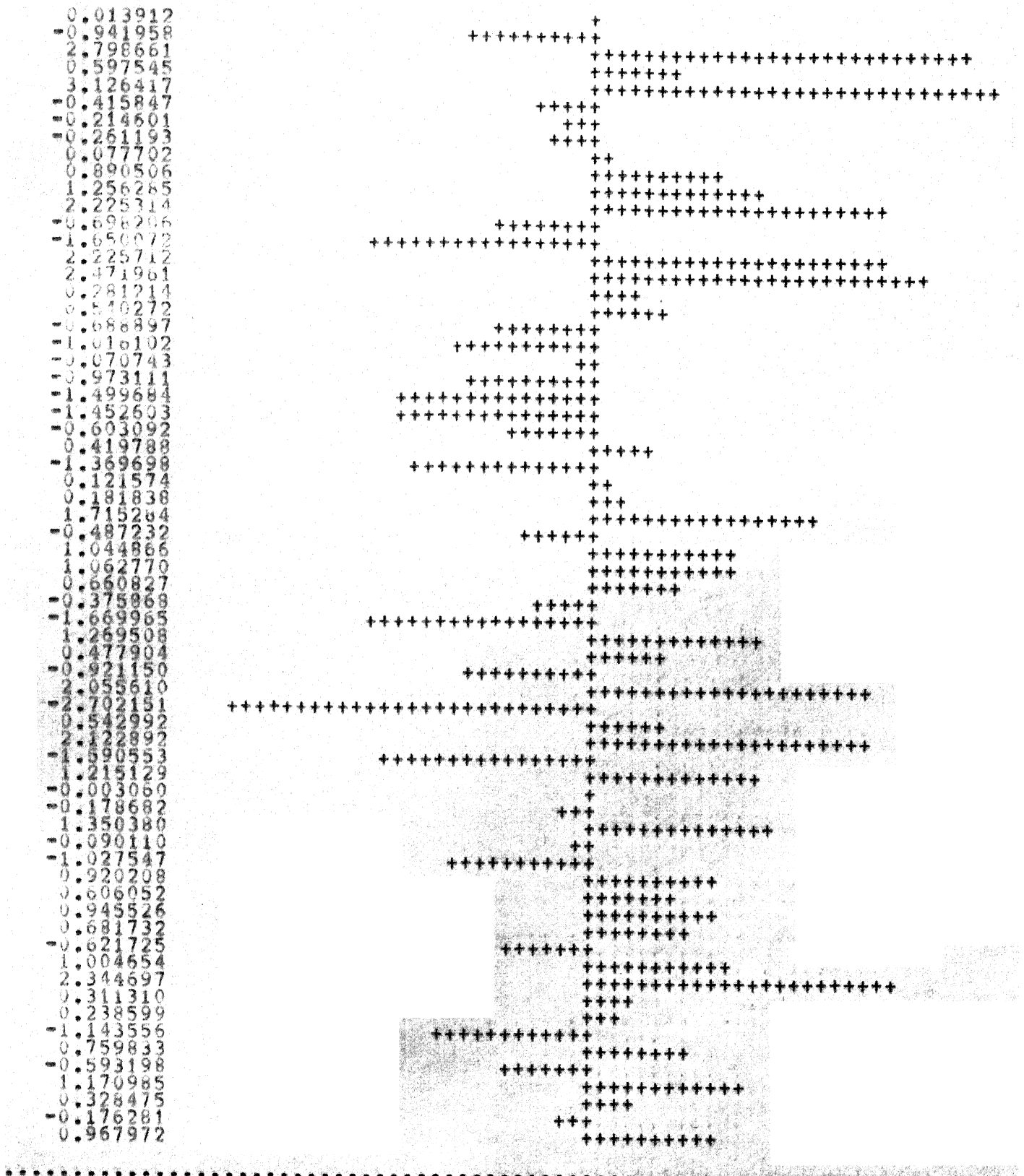


Fig. 4.2-3 : Primary noise $n_d(j)$ [without bias].

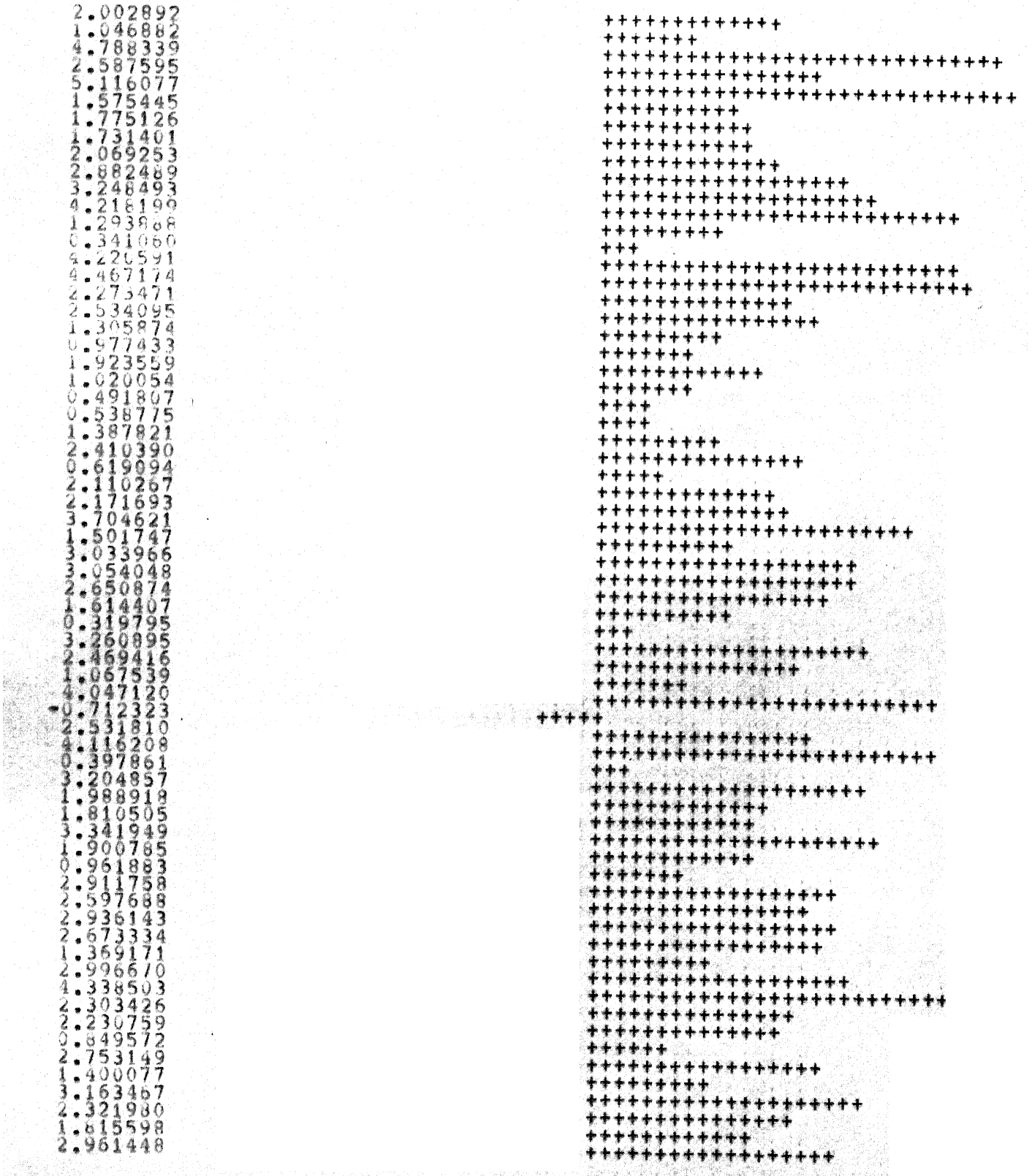


Fig. 4.2-4 : Noise estimate y(j).

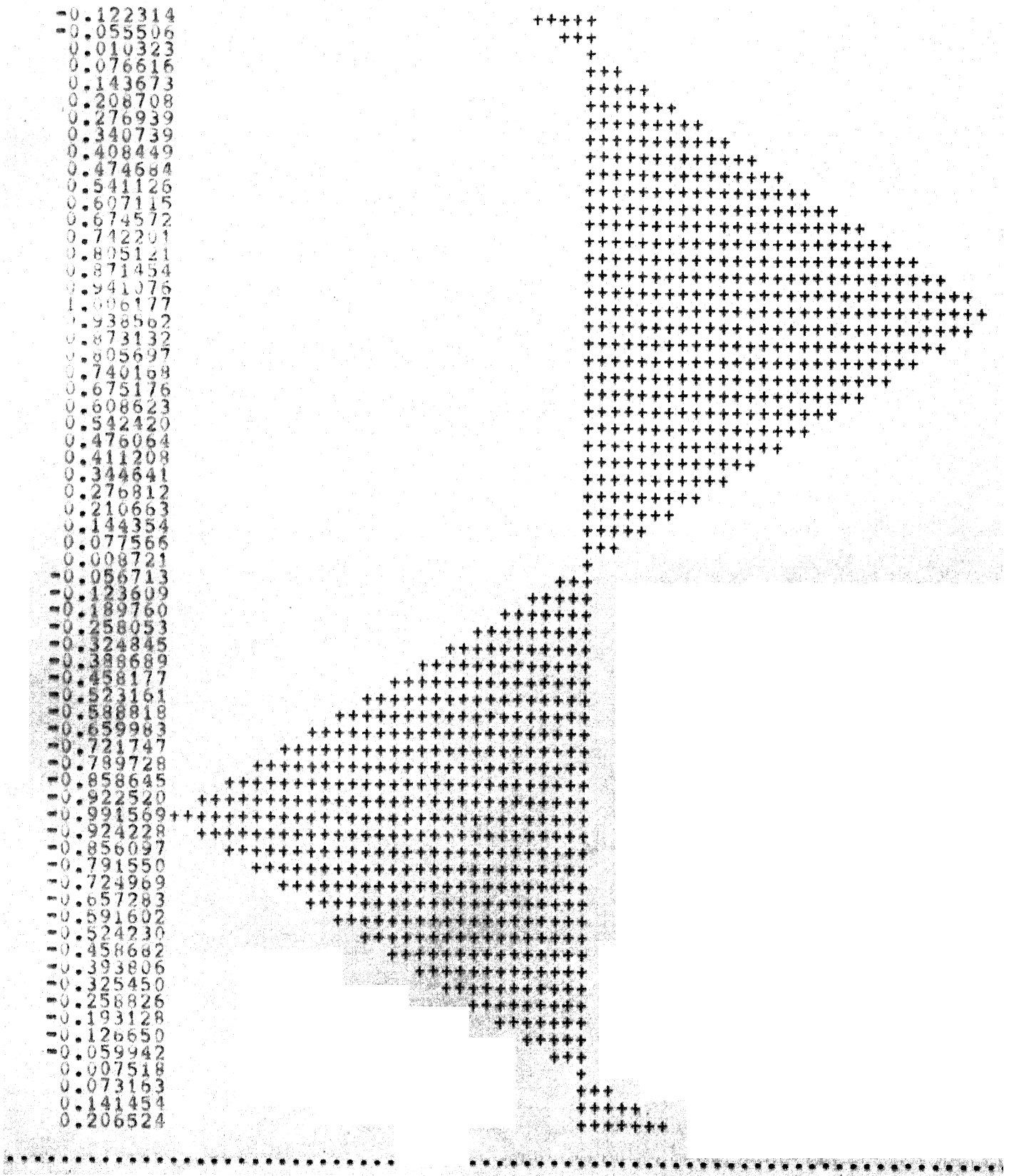


Fig. 4.2-5 : ANC output z(i).

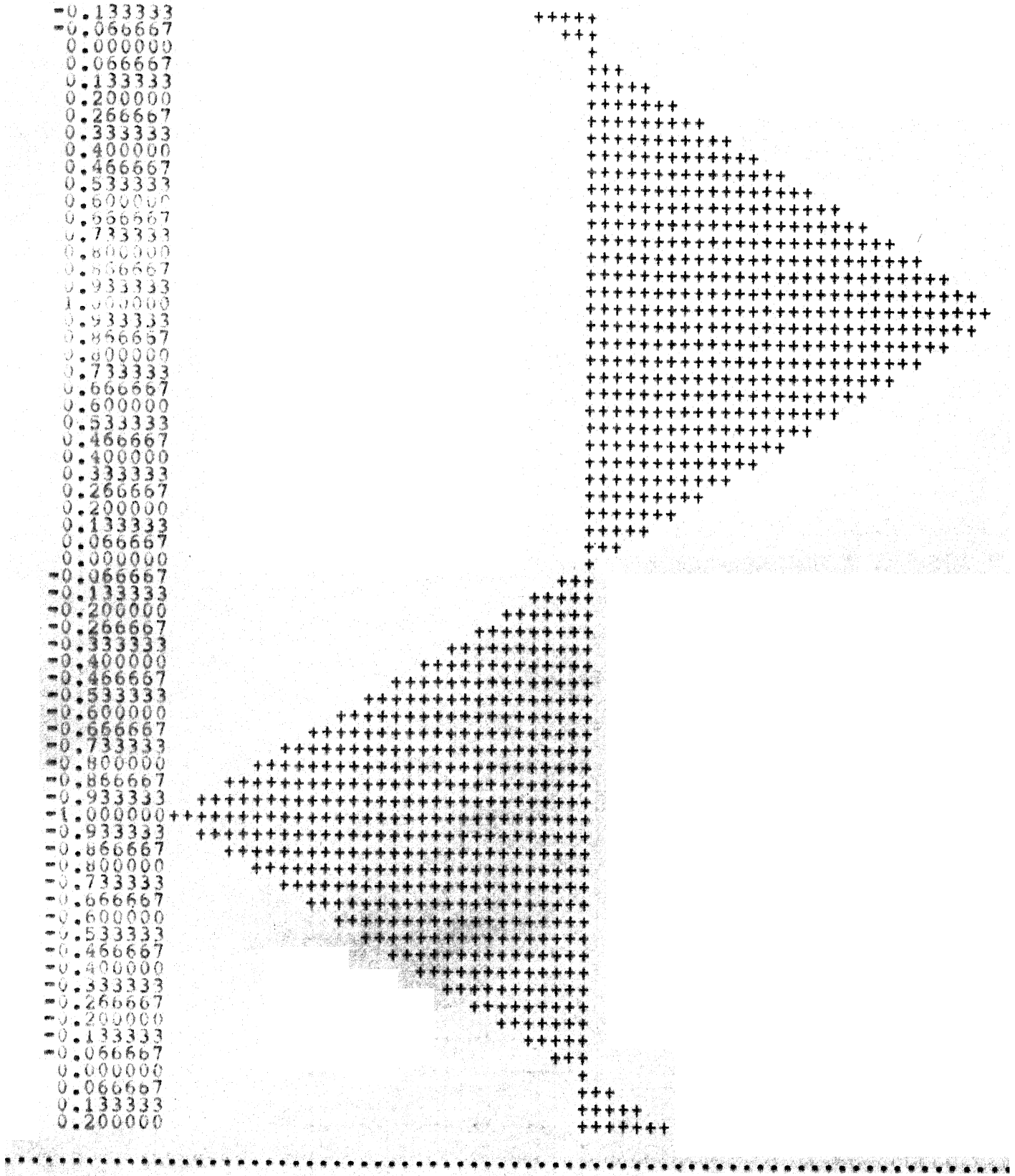
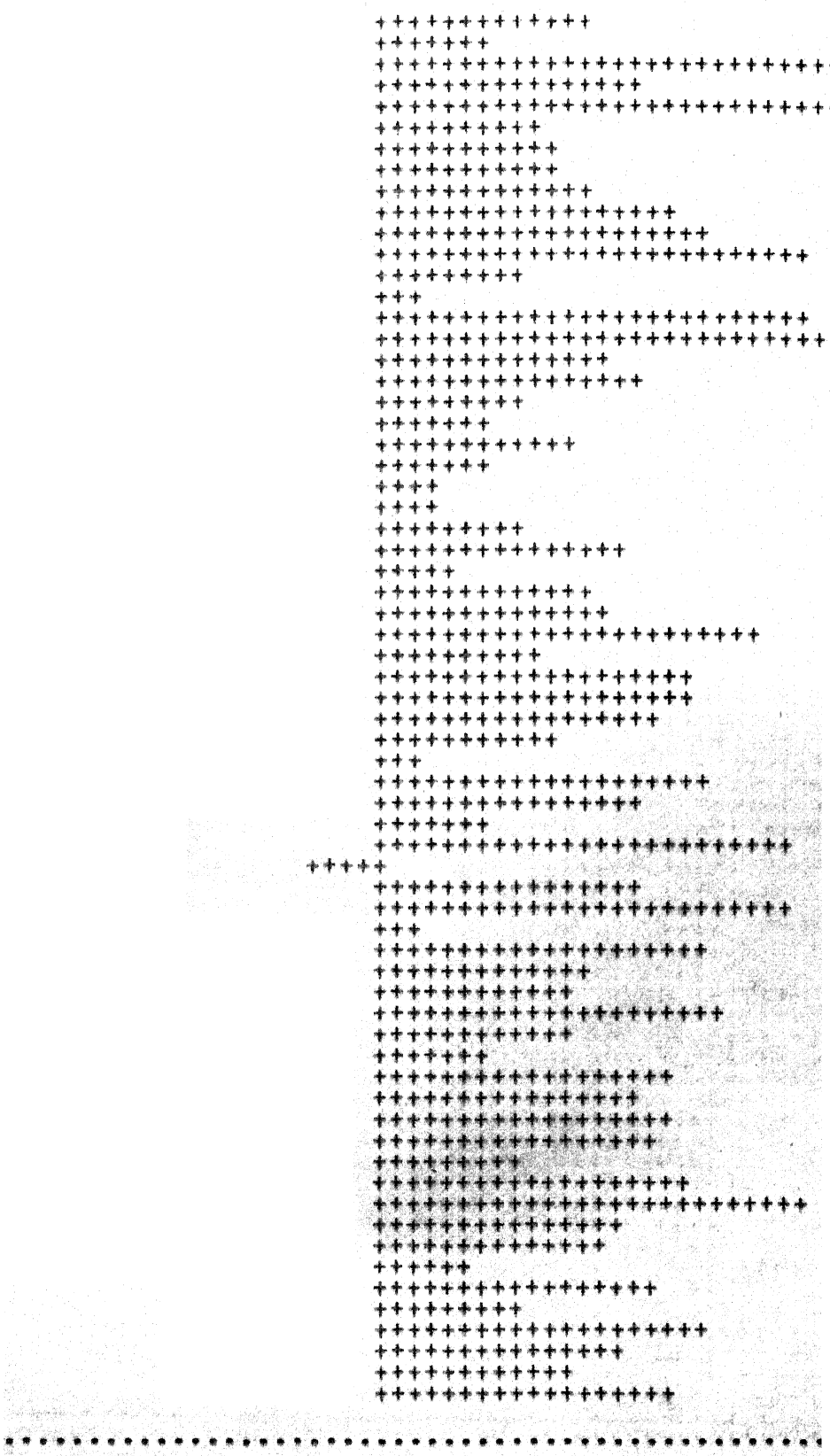


Fig. 4.2-6 : Signal s(t).



Primary noise with bias.

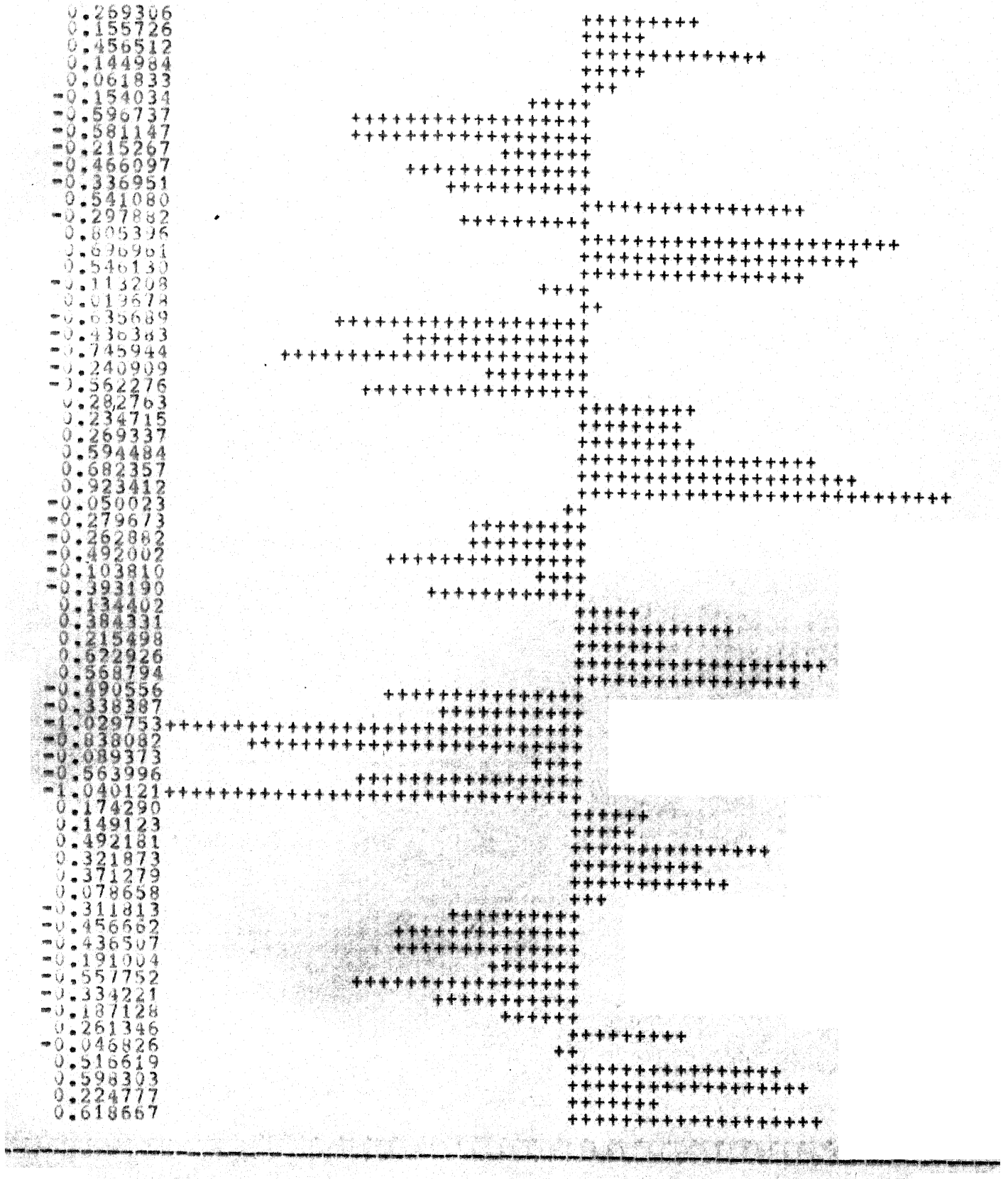


Fig. 4.3-1 : Primary input d(k) to the adaptive self-tuner.

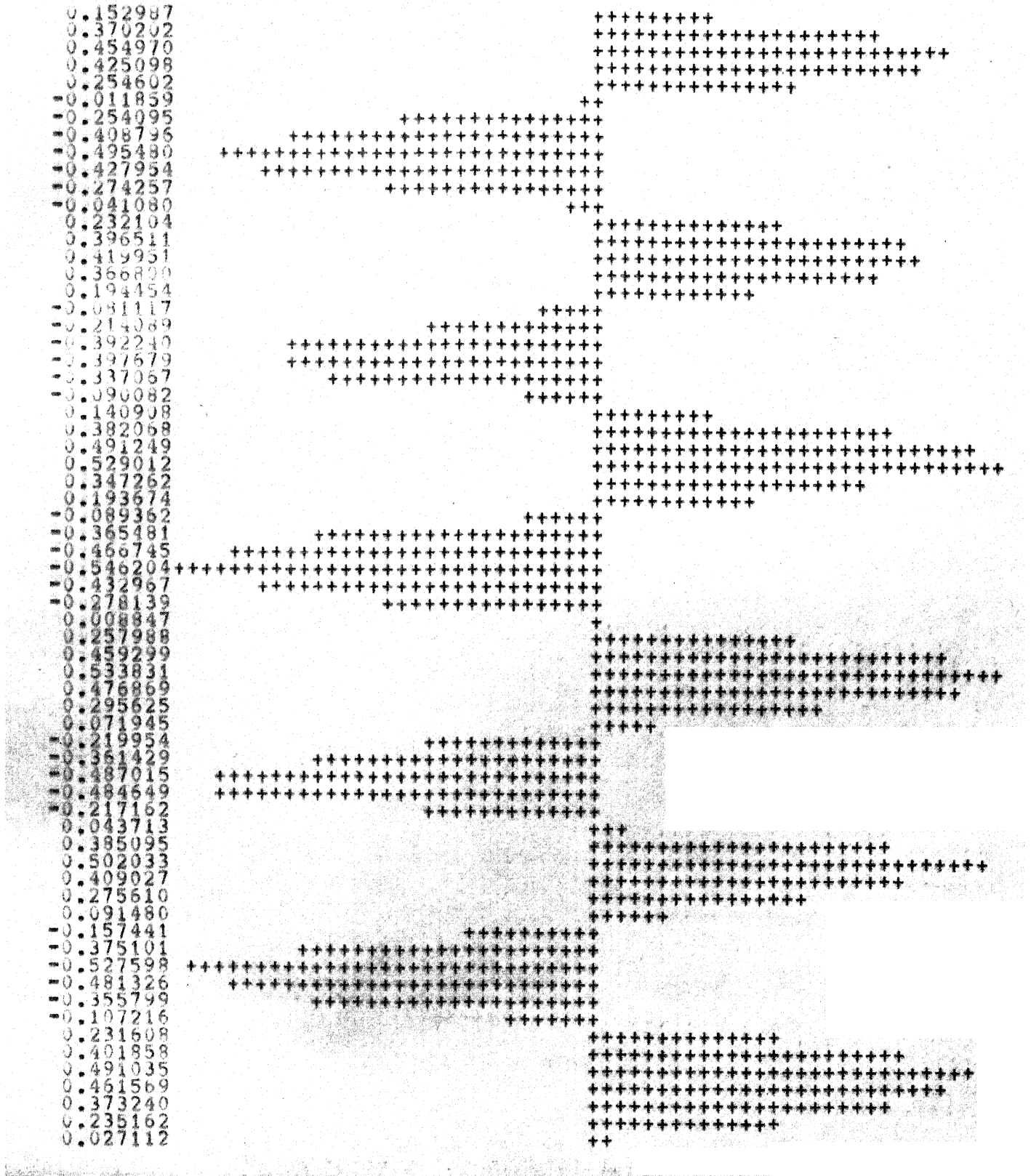


Fig. 4.3-2 : Self-tuner output y(j).

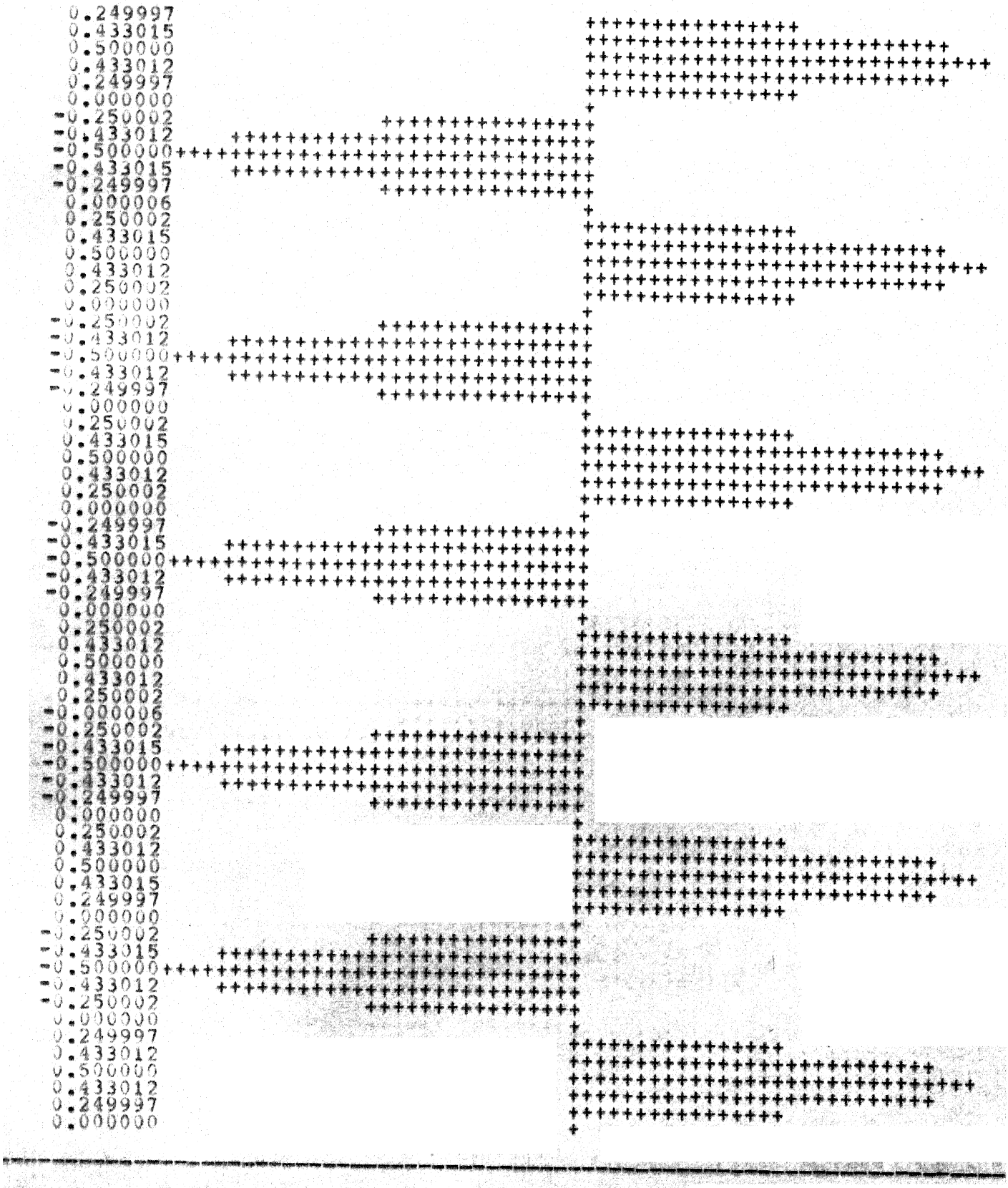


Fig. 4.3-3 : Signal s(t).

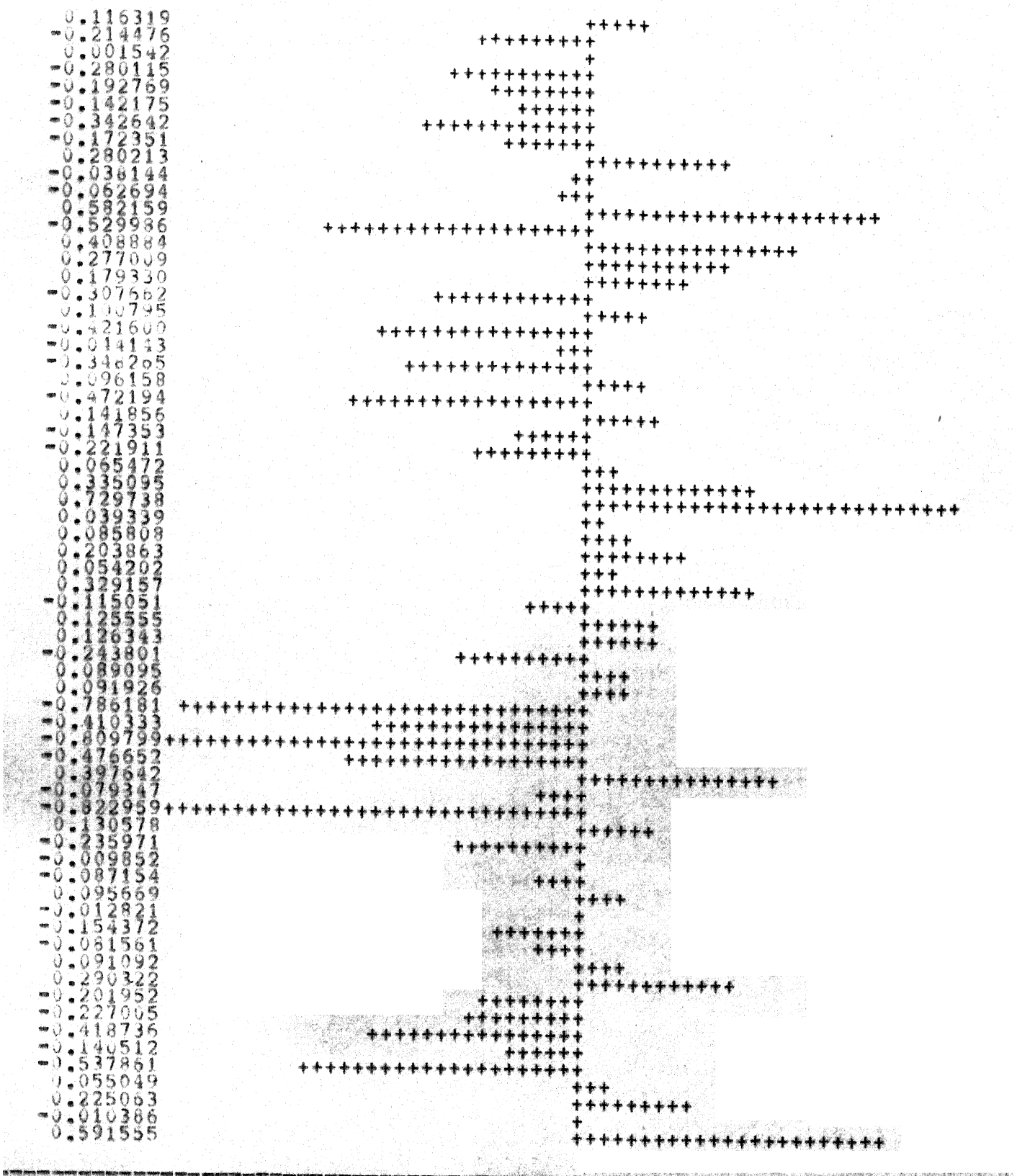


Fig. 4.3-4 : ANC output z(j).

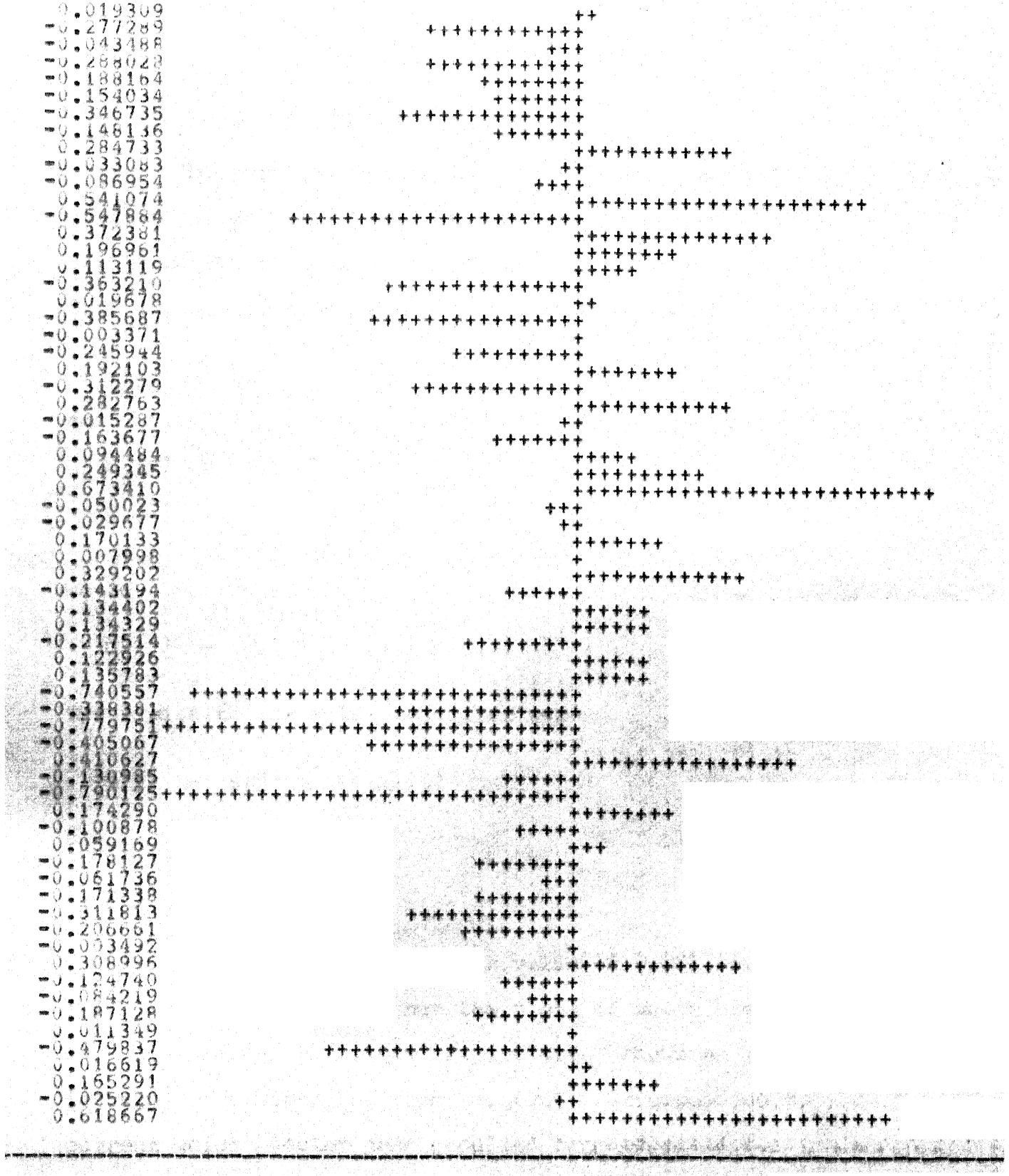


Fig. 4.3-5 : Broad-band noise x(i).

4.4 The ANC with Stationary Inputs

The purpose of this treatment is to illustrate the various design rules given in Section 2. Two independent zero-mean white Gaussian sequences, q(j) and n(j), of powers 0.01 and 2.0, respectively, were combined to form the signals

$$s(j) = 0.99\, s(j-1) + q(j)$$

and

$$n_x(j) = 0.6\, n_x(j-1) + 0.2\, n_x(j-2) - 0.3\, n_x(j-3) - 0.1\, n_x(j-4) + 0.7\, n_d(j)$$

where

$$n_d(j) = 0.1\, n_d(j-1) + n(j) .$$

The inputs to the noise canceller were

$$d(j) = s(j) + 1.4\, n_d(j)$$

and

$$x(j) = n_x(j) .$$

Using five weights and a value of 0.001 for μ, a simulation was carried out the plots of which are shown in Figures 4.4-1 to 4.4-6. Figure 4.4-7 shows an individual learning curve obtained from Eq. (2.7) corresponding to the current weight vector that resulted from adaptation. Figure 4.4-8 shows a somewhat smooth learning curve obtained by averaging over an ensemble of 120 simulations, each

starting with the same initial vector $\underline{W}(0) = \underline{0}$ and each having a different input derived from the same "statistical population." The interval between successive ordinates in both the plots is 10 iterations. The theoretical results and the results obtained from simulation are presented and compared in what follows.

The estimates of the correlation matrices are:

$$R_x = \begin{bmatrix} 4.2769 & 2.8866 & 1.5657 & -0.0529 & -1.0076 \\ 2.8867 & 4.2448 & 2.8653 & 1.5602 & -0.0617 \\ 1.5658 & 2.8653 & 4.2968 & 2.9156 & 1.6076 \\ -0.0529 & 1.5602 & 2.9156 & 4.2752 & 2.8839 \\ -1.0076 & -0.0617 & 1.6076 & 2.8834 & 4.2615 \end{bmatrix}$$

and

$$\underline{R}_{dx} = \begin{bmatrix} 4.2308 \\ 0.4578 \\ 0.0461 \\ -0.0025 \\ -0.0015 \end{bmatrix} .$$

The theoretical weights obtained from Eq. (2.8) and the simulated weights measured at different instants are given in Table 4.3. The eigenvalues of R_x are given by

$$\underline{\Lambda} = \begin{bmatrix} 10.9182 \\ 7.2007 \\ 0.7758 \\ 1.5683 \\ 0.8920 \end{bmatrix} .$$

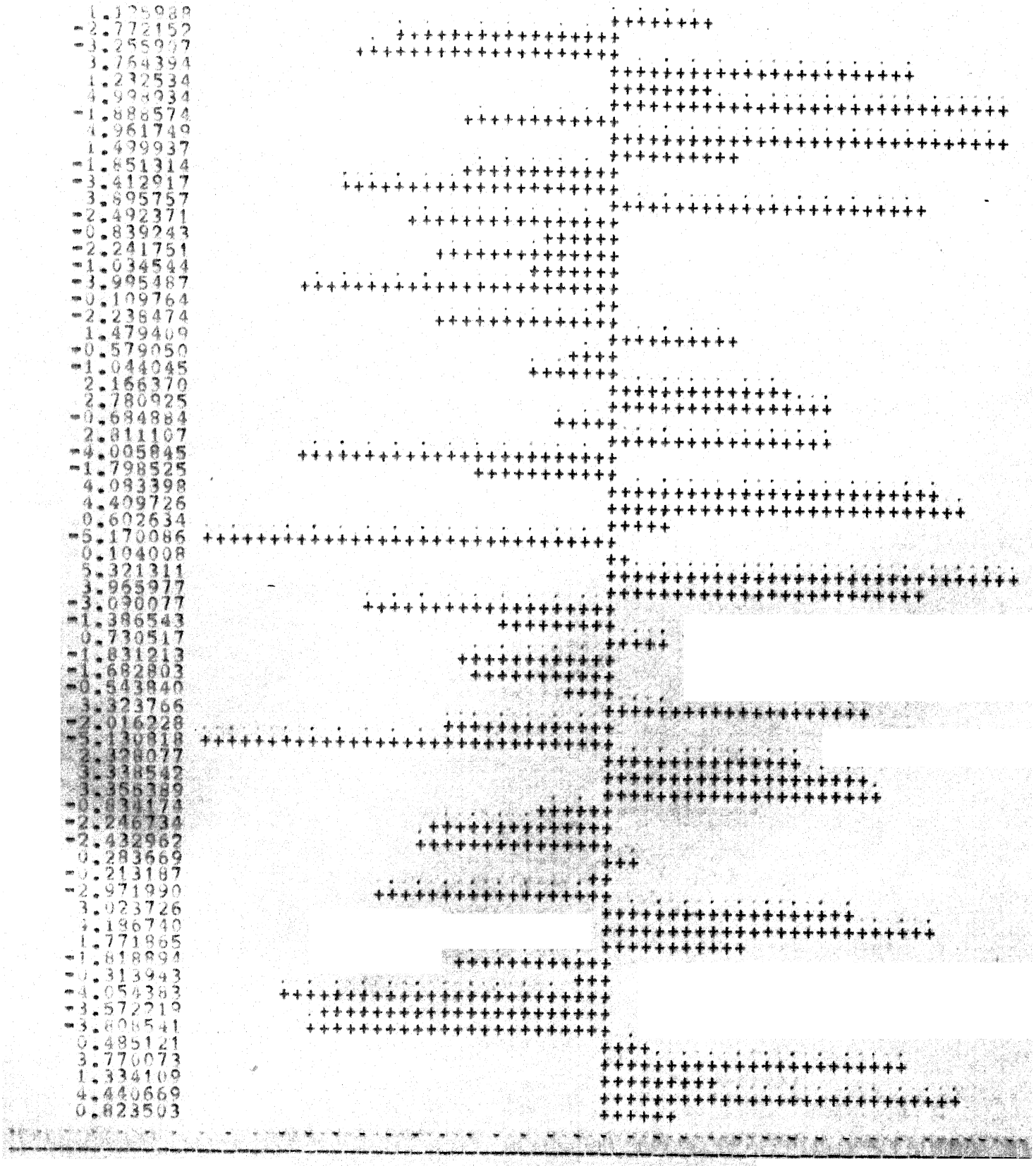


Fig. 4.4-1 : Primary input d(i).

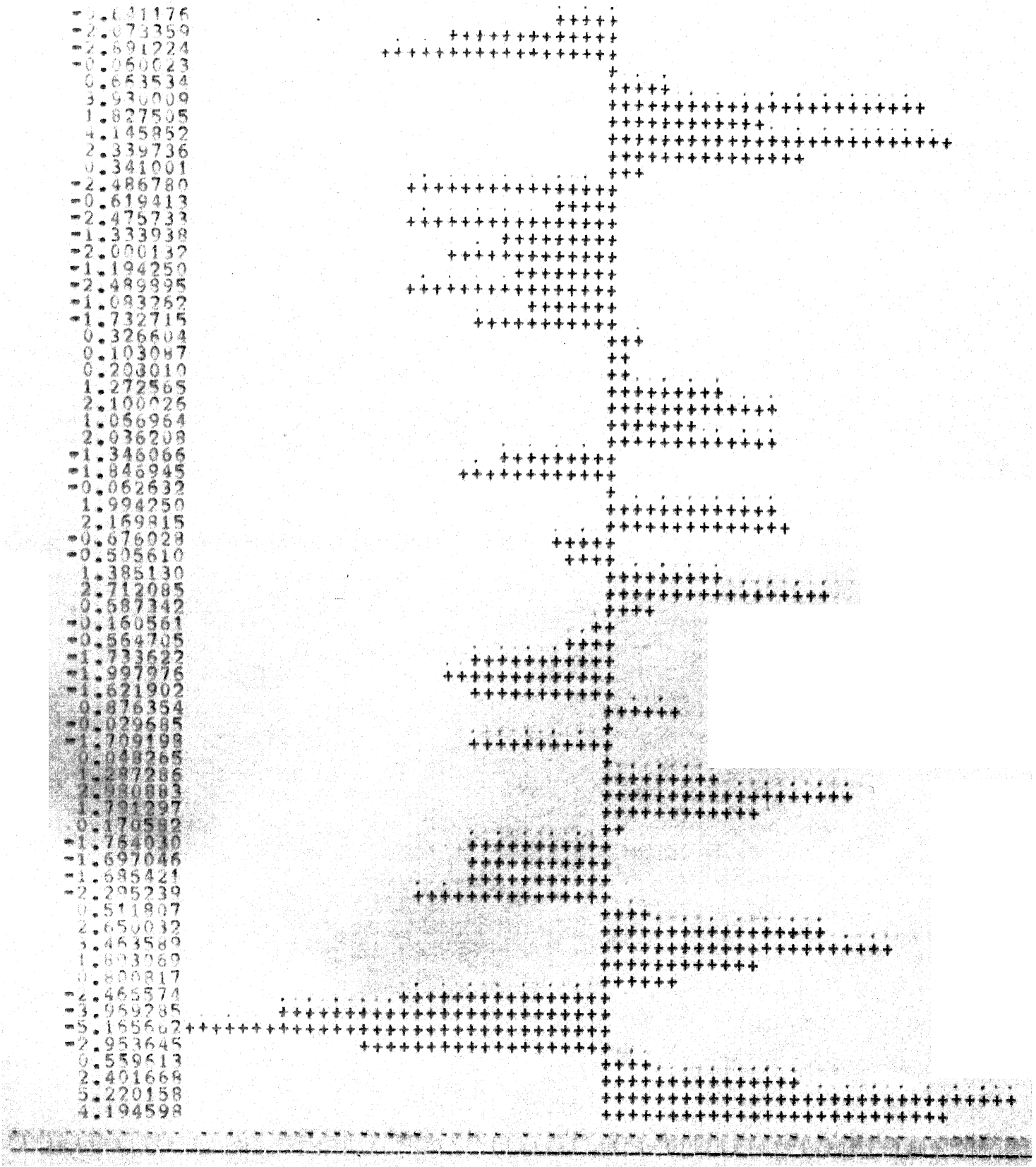


Fig. 4.4-2 : Reference input x(i).

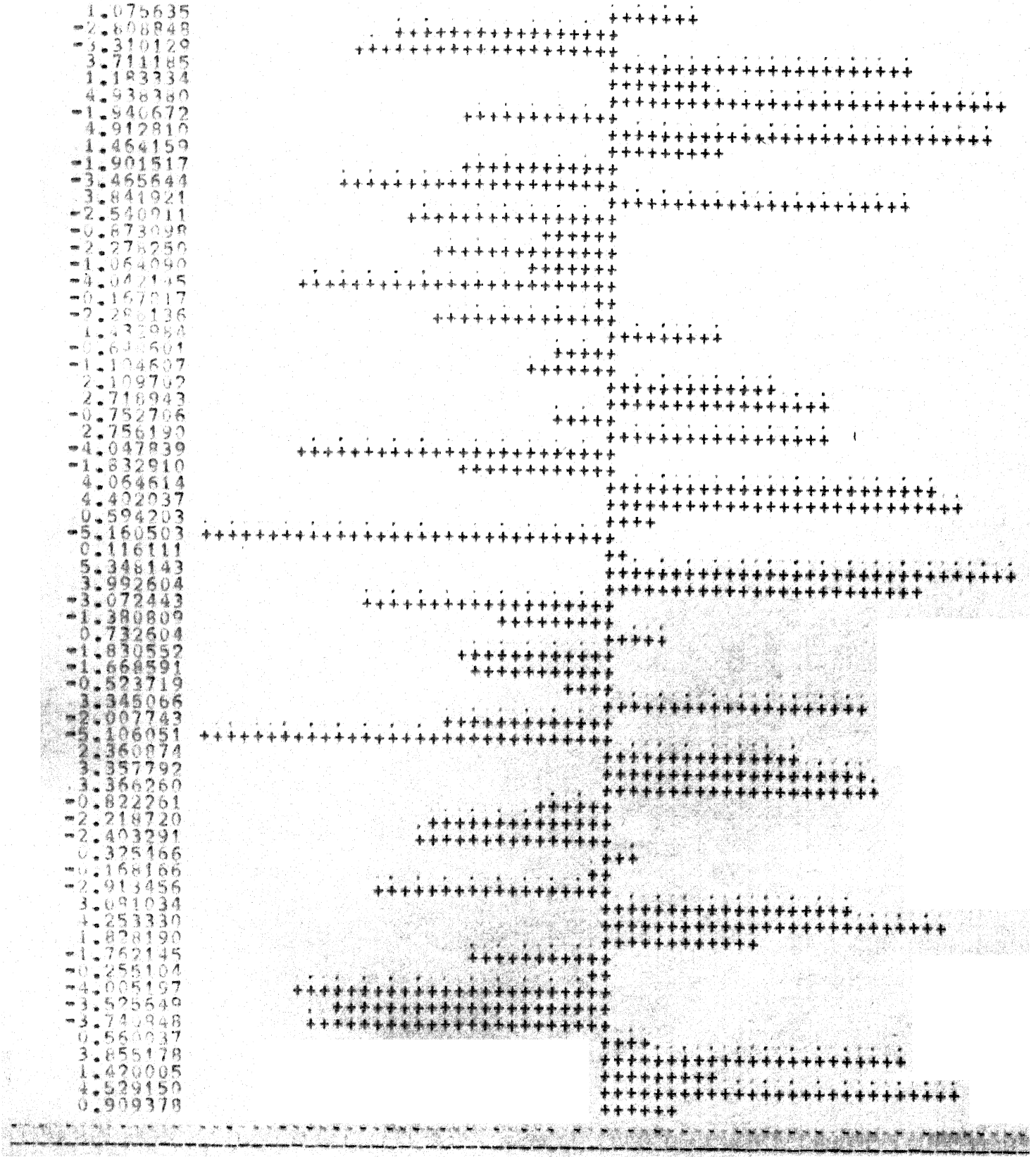


Fig. 4.4-3 : Primary noise $n_d(i)$.

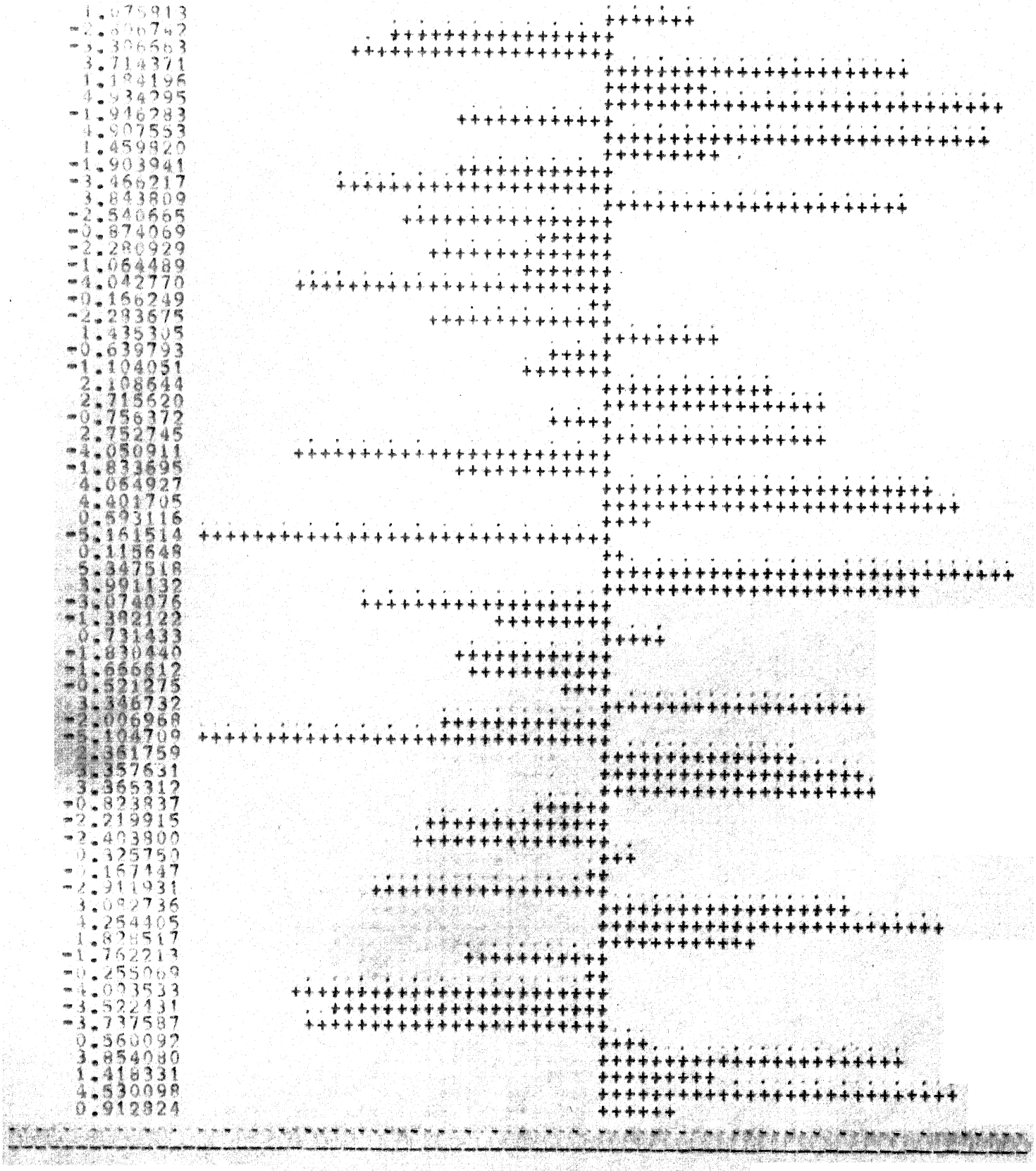


Fig. 4.4-4 : Noise estimate v(4).

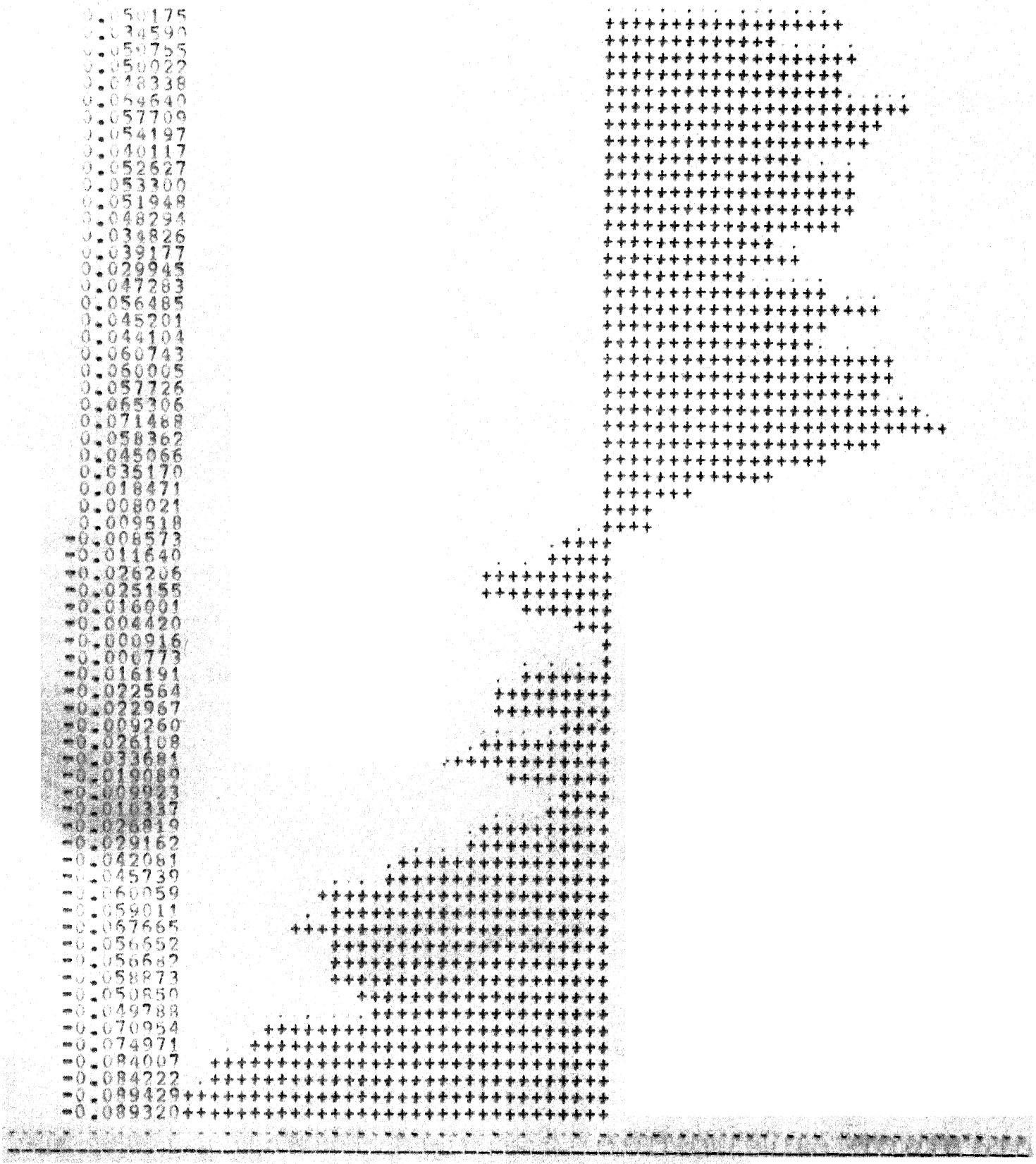


Fig. 4.4-5 : Signal estimate $\hat{s}(n)$.

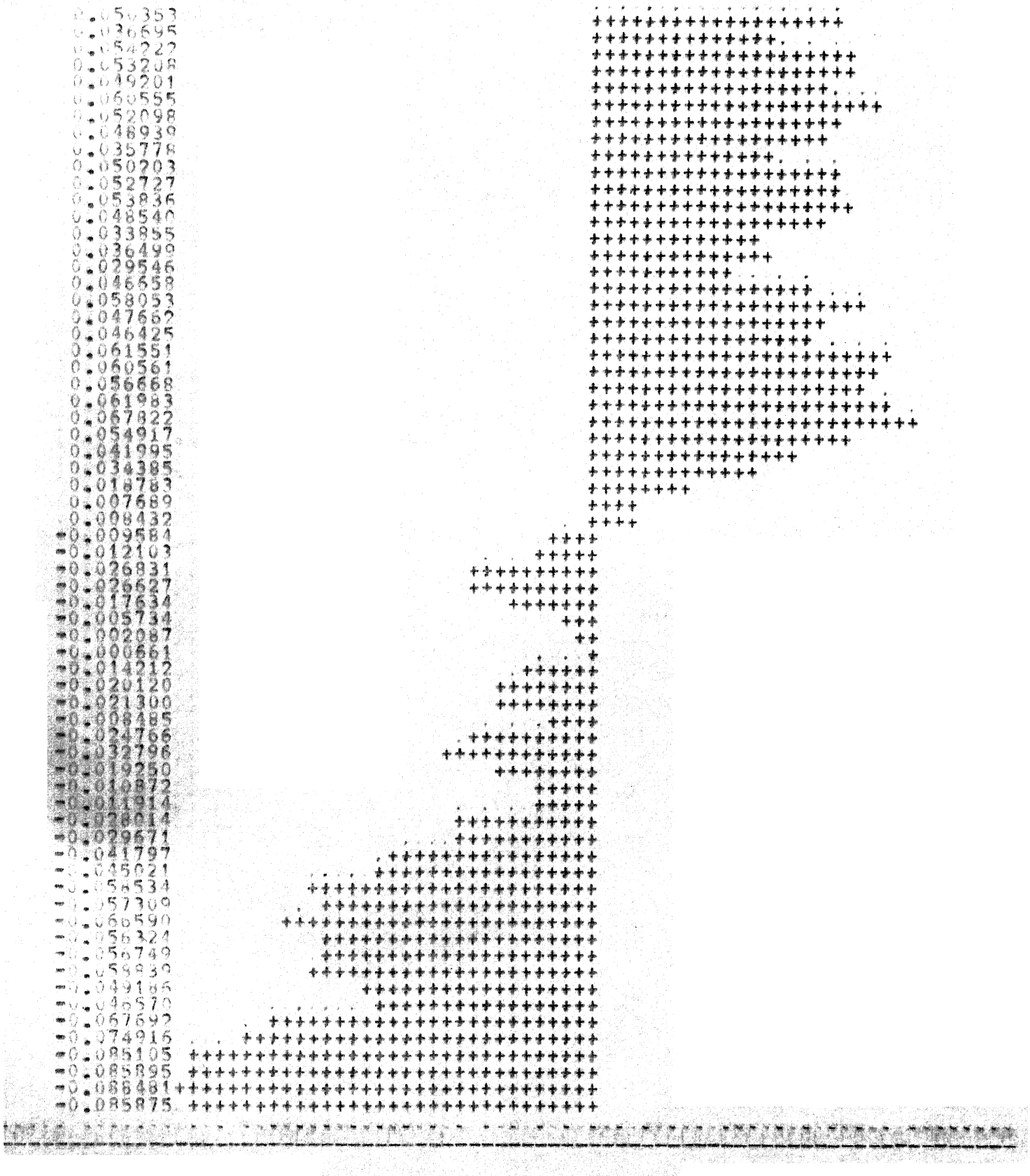


Fig. 4.4-6 : Signal s(i).

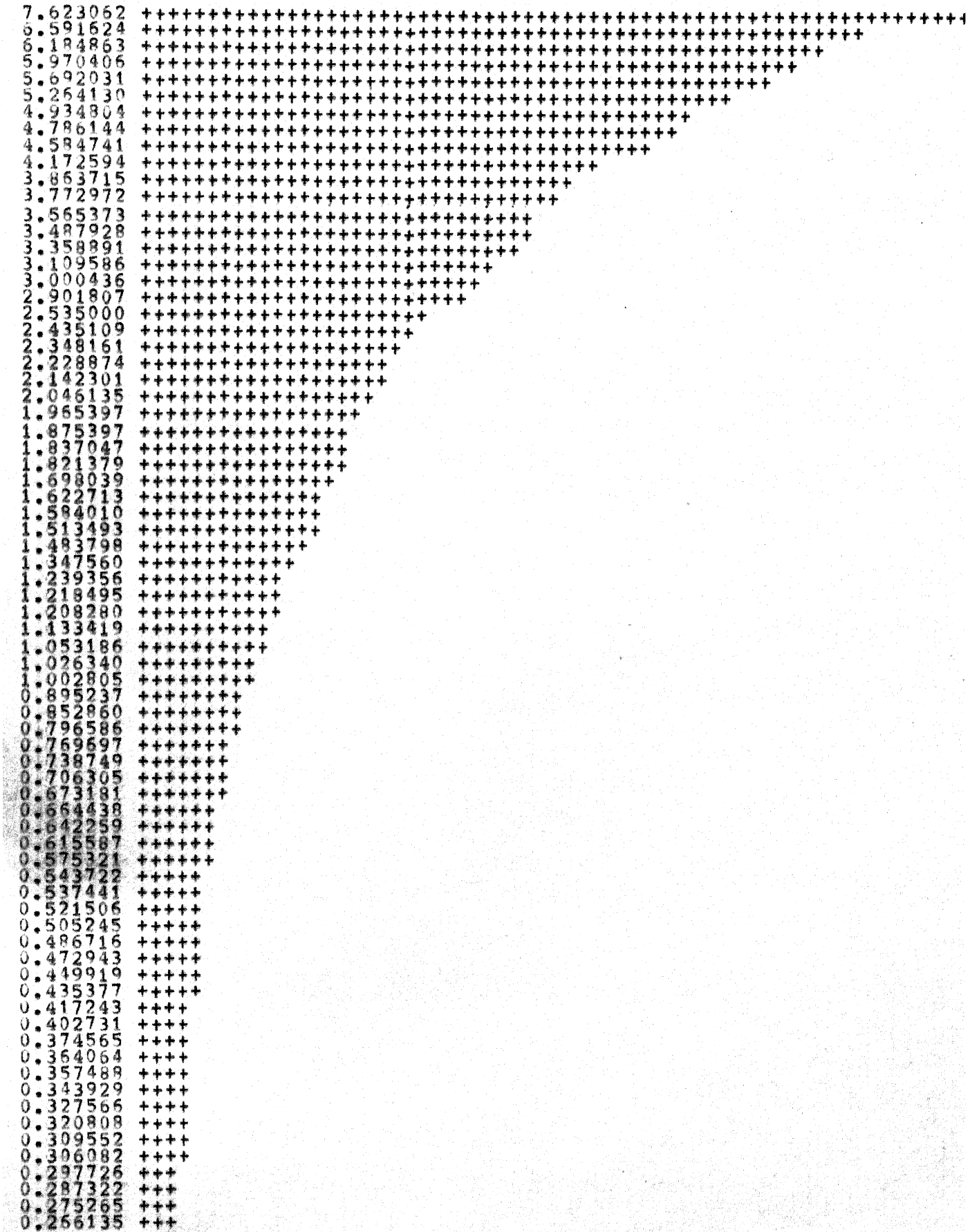


Figure 4.4-7 : Individual learning curve.

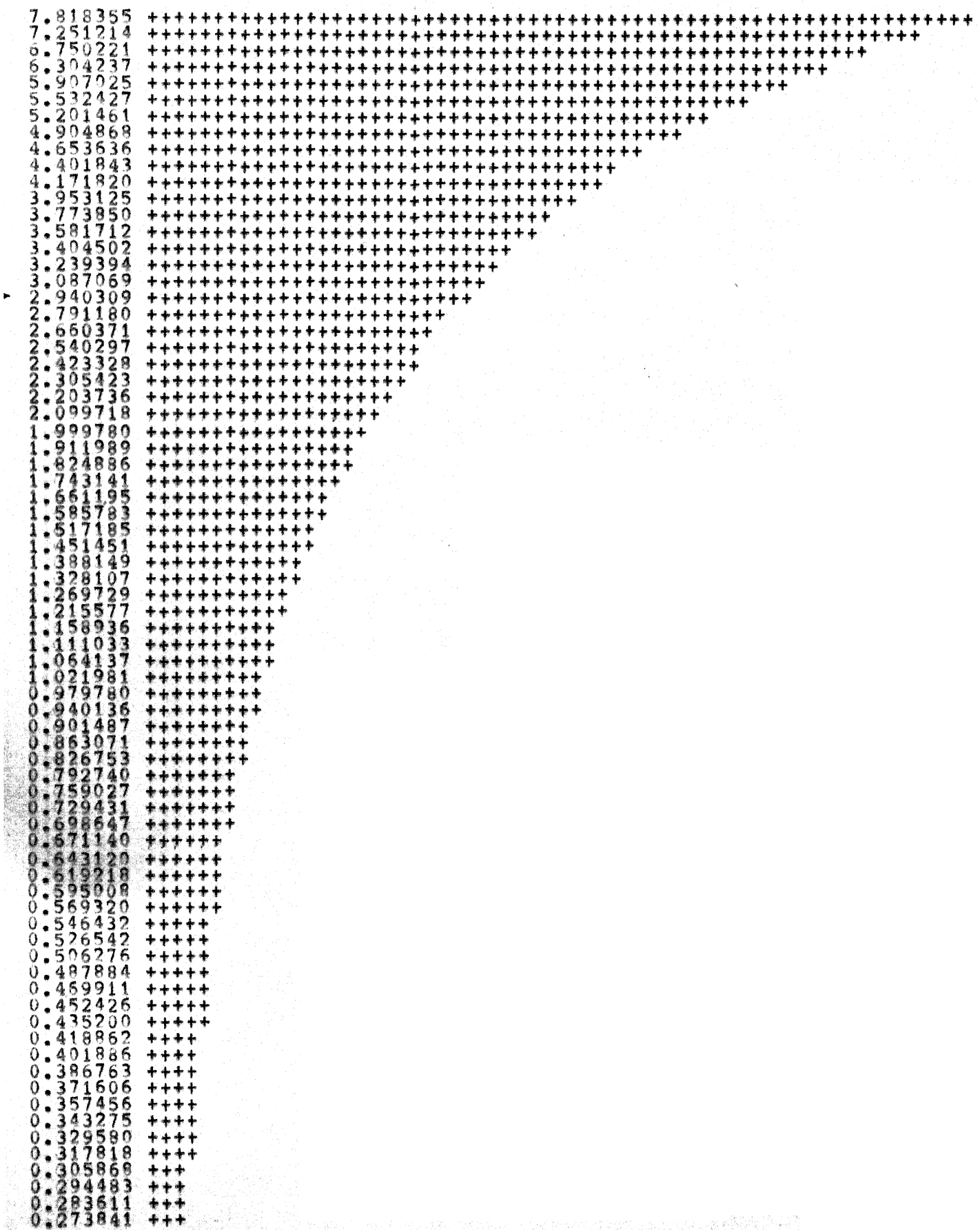


Figure 4.4-8 : Ensemble learning curve.

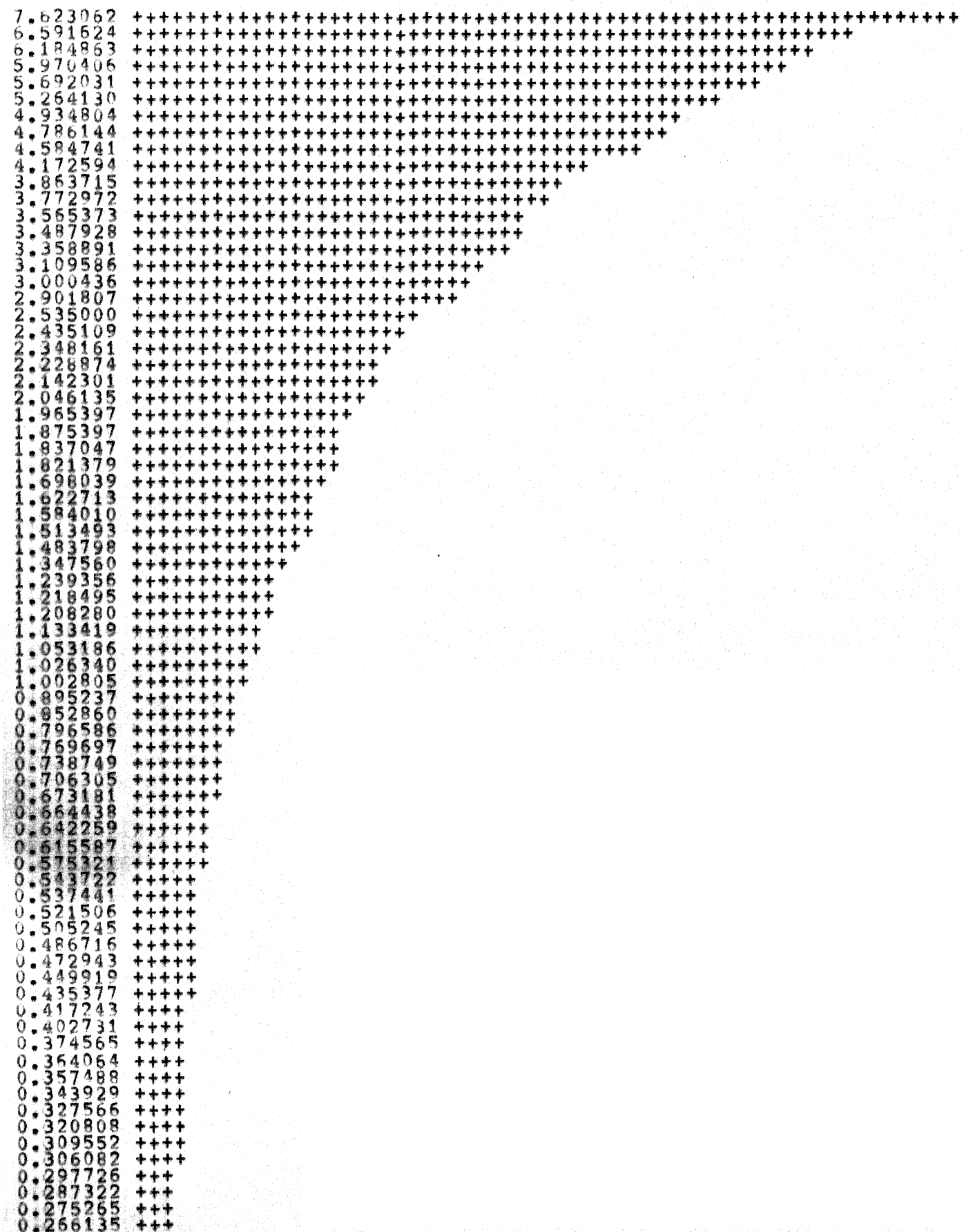


Figure 4.4-7 : Individual learning curve.

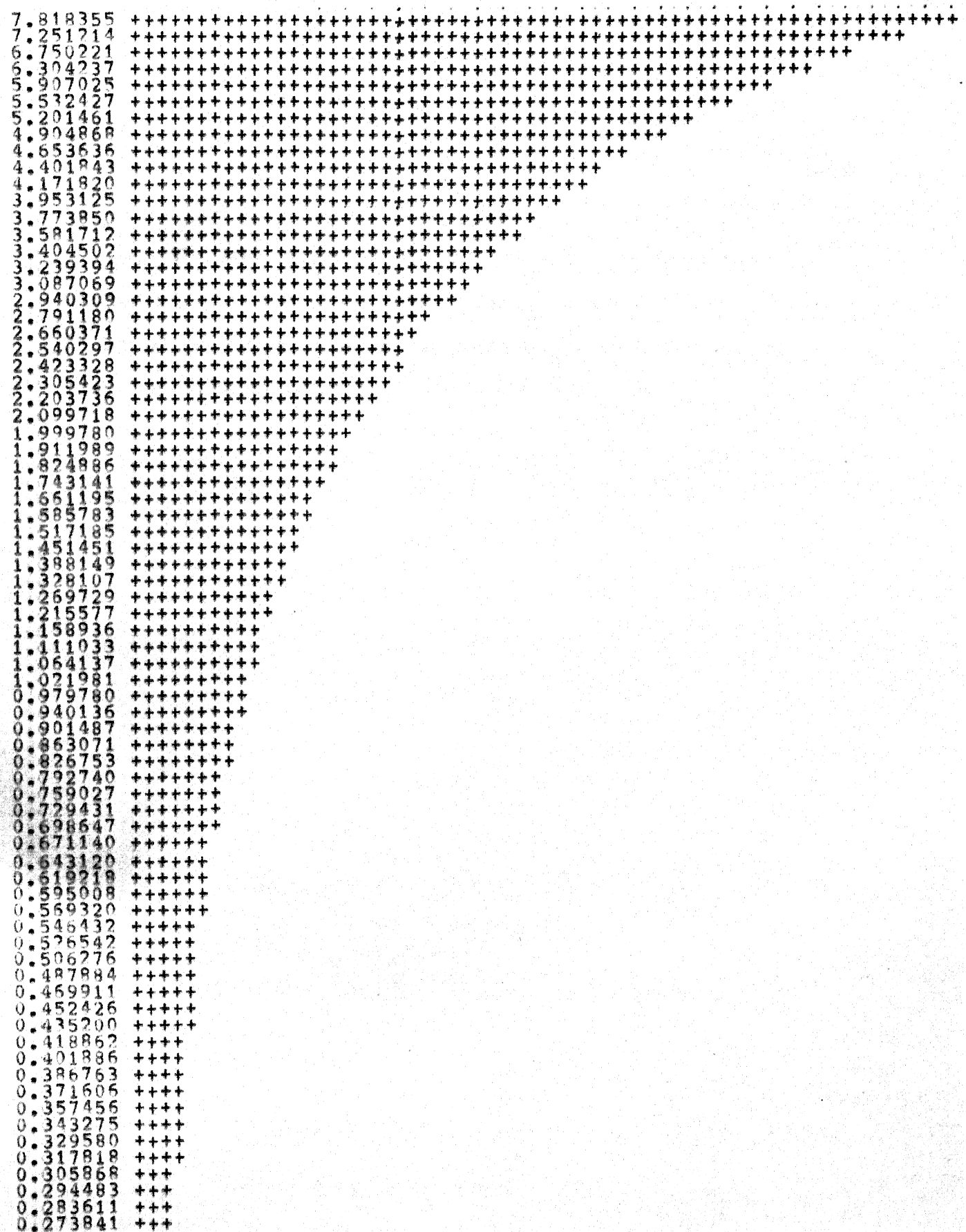


Figure 4.4-8 : Ensemble learning curve.

The ANC produced about 2.5% more mse than the Wiener filter; but then, the Wiener filter could have been designed only with the complete knowledge of the second-order statistics of the signal and noise.

4.5 The ANC with Nonstationary Inputs

Some of the best applications of ANC's will doubtless be made to the filtering of nonstationary inputs. When a slowly-varying process is applied to an ANC, the latter tracks the characteristics of the former in a sub-optimum way [2-2]. Means of describing the behaviour of adaptive systems with nonstationary inputs is currently a subject of intensive research.

Two simulations were carried out to demonstrate the effectiveness of ANC's with nonstationary inputs. The same signals and processing parameters, used in the previous simulation, were used in the present ones, but for the differences as stated in what follows.

In the first simulation, the variance of n(j) was modulated by a square-wave with period T τ_{mse}. Once the initial transients were over, the adaptive filter was seen to track the signal characteristics with little change.

In the second simulation, the variance of q(j) was stepped up by 20 dB and 30 dB after the steady-state had been reached. We shall define the convergence time as the time required for the filter weights to settle within 5% of their steady-state values. The convergence times of the weights for different μ's and signal power levels are tabulated in page 4-44 (Table 4.4).

Next, the channel coefficients were slowly varied. The plots of one of the channel coefficients and the

corresponding filter weight that tracks it are given in Figures 4.5-1 and 4.5-2, respectively. It is seen that the weight tracks the channel variations in a sub-optimum way.

Although for simplicity the zeros of the transfer function of the channel models considered in this section have been assumed absent, their introduction will increase the number of filter weight considerably. A good approximation of a single pole filter by an FIR filter, for instance, requires as many as 10 filter weights [2-9]; on the other hand, an IIR filter will require just one weight.

Table 4.4 : Convergence time for nonstationary inputs.

Signal power change	μ	Settling time (No. of iterations)
20 dB	0.005	243
	0.001	297
	0.0005	356
30 dB	0.001	365
	0.0004	407
	0.0007	458

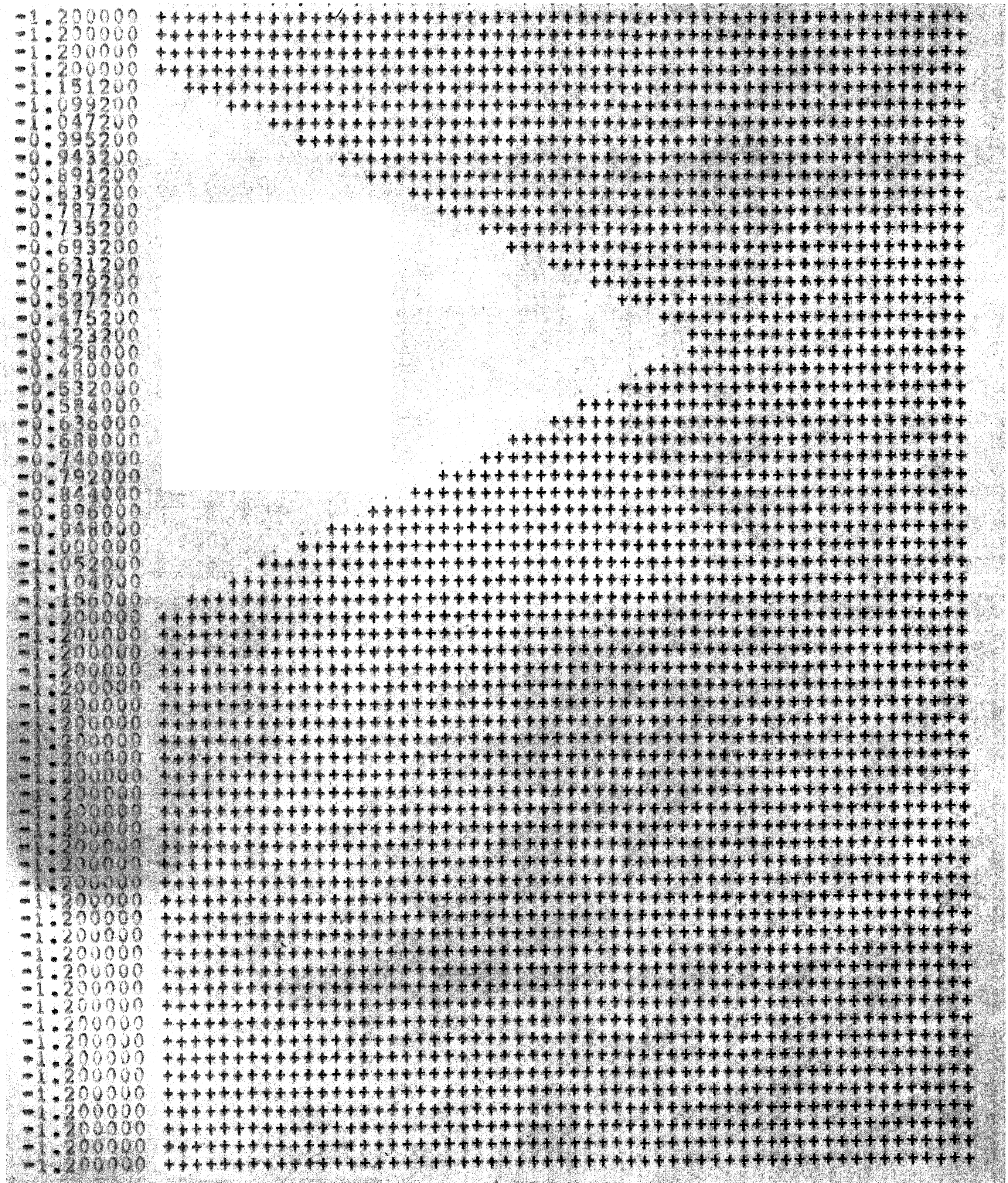


Fig. 4.5-1 : Variation of channel parameter with time.

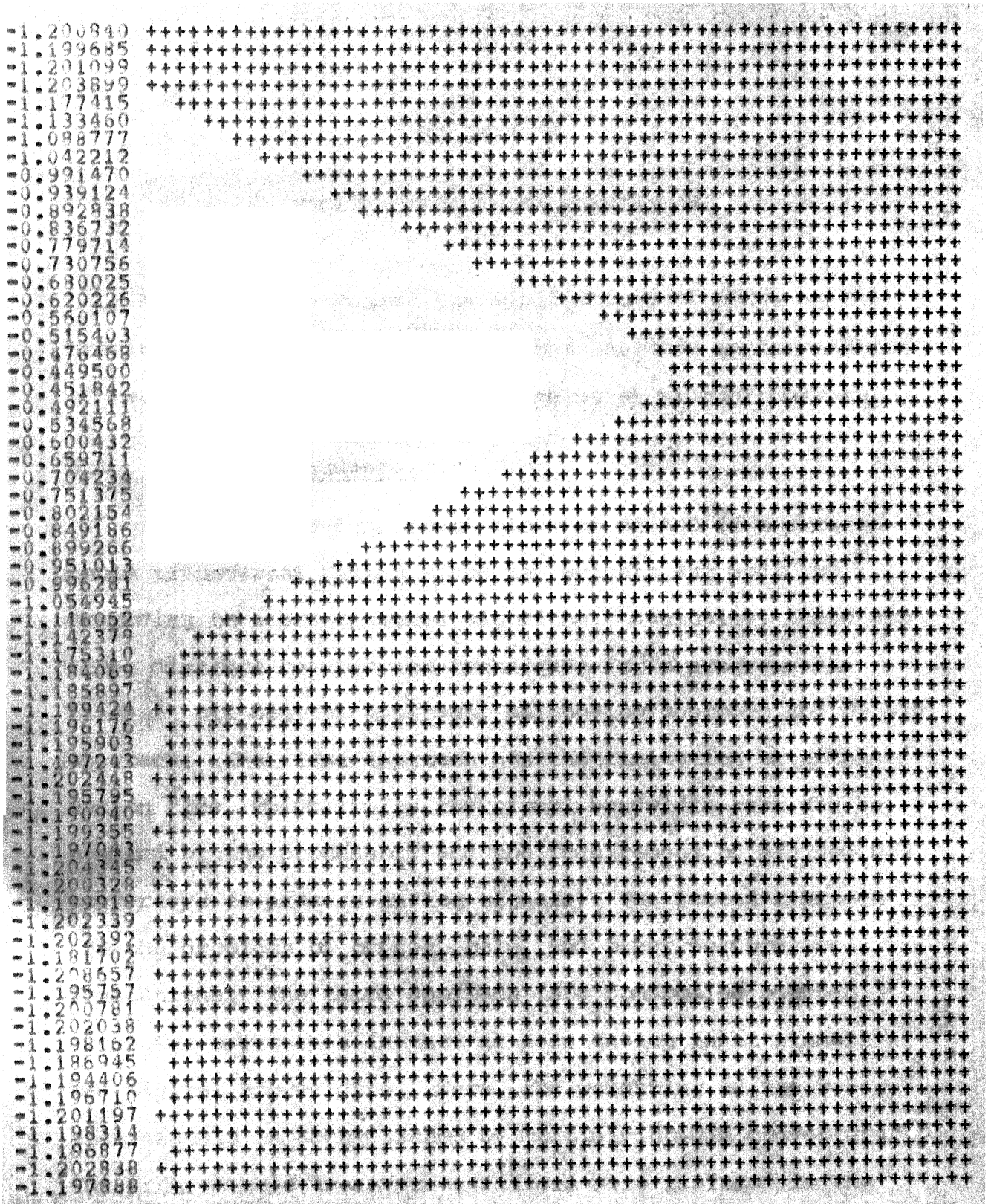


Fig. 4.5-2 : Variation of filter weight with time.

SECTION 5

MICROPROCESSOR IMPLEMENTATION

Having discussed the applications of ANC's in the previous section, we now examine the hardware implementation of real-time noise cancellers by means of microprocessors.

5.1 Choice of Hardware

The basic functional block of an ANC is a programmable transversal filter whose tap weights are modified according to a pre-assigned algorithm. Basically, there are three distinct realization techniques for a programmable filter: all digital approach, all analog approach and hybrid approach. The first approach has the limitation of computation time, which limits the signal bandwidth that can be handled by the processor, and the requirement of A/D and D/A converters to process analog signals. The second approach has the problems of offset, gain, and other performance limitations. The third approach is a compromise between these two approaches in that it uses analog input signals and digital tap weights. Here, the weighting of the reference signals is accomplished by multiplying-D/A converters. The programmable transversal filters were realized in the past by means of hard-wired logic circuits and analog components. Recently, single chip CCD adaptive filters [2-5]

Table 5.1 : Salient features of microprocessors.

Processor	INTEL 8080A	INTEL 8086	TI 9900	ZILOG 8000	ISI 11
Technology	NMOS	NMOS	NMOS	NMOS	NMOS
Word size data/instruction	8/8	16/16	16/16	16/16	16/16
Number of instructions	78	97	69	110	100
Maximum clock frequency/phase	2.6/2	5/1	4/4	8/1	-
Instruction time shortest/longest	1.5/3.75	0.4/37	2/31	0.75/90	-
TTL compatible	YES	YES	YES	YES	YES
BCD arithmetic	YES	YES	NO	YES	NO
On-chip interrupt levels	1	1	16	1	1
Number of general purpose registers	8	8	16	16	8
DMA capability	YES	YES	YES	YES	YES
Prototype system availability	YES	YES	NO	YES	YES
Package size (pins)	40	40	64	40/48	SBC
Voltage required (volts)	5, 12 -5	5	5, 12 -5	5	5
Assembly language development system	YES	YES	YES	YES	YES

The Zilog 8000 excels the other processors in respect of instruction cycle time, instruction set and number of registers. For this project, an Intel 8080 processor was chosen because of its ready availability and the standard support software available with it.

5.2 Implementation Details

We now need to examine which algorithm to choose for implementation. Among the algorithms discussed in Section 2, the LMS algorithm and the CLMS algorithm are amenable for microprocessor implementation. The latter algorithm involves less "number crunching" on the processor, though it converges slower than the former by a factor of $\pi/2$. In order to facilitate signal processing in real-time, the computational burden on the processor must be minimized. So, the CLMS algorithm was chosen for implementation.

The flowchart for the CLMS algorithm is shown in Figure 5.1. An assembly language program was developed and tested on ECIL's microcomputer, MICRO-78. The computer has been built around an Intel 8080 processor, and has, among other facilities, a resident assembler and an on-line debugging system (ODS). Description of the MICRO-78 and the Intel 8080 processor may be found in [5-4].

To minimize the errors arising from the digital implementation, multiple-precision arithmetic has been used. The I/O data are represented by 9 bits (this will correspond

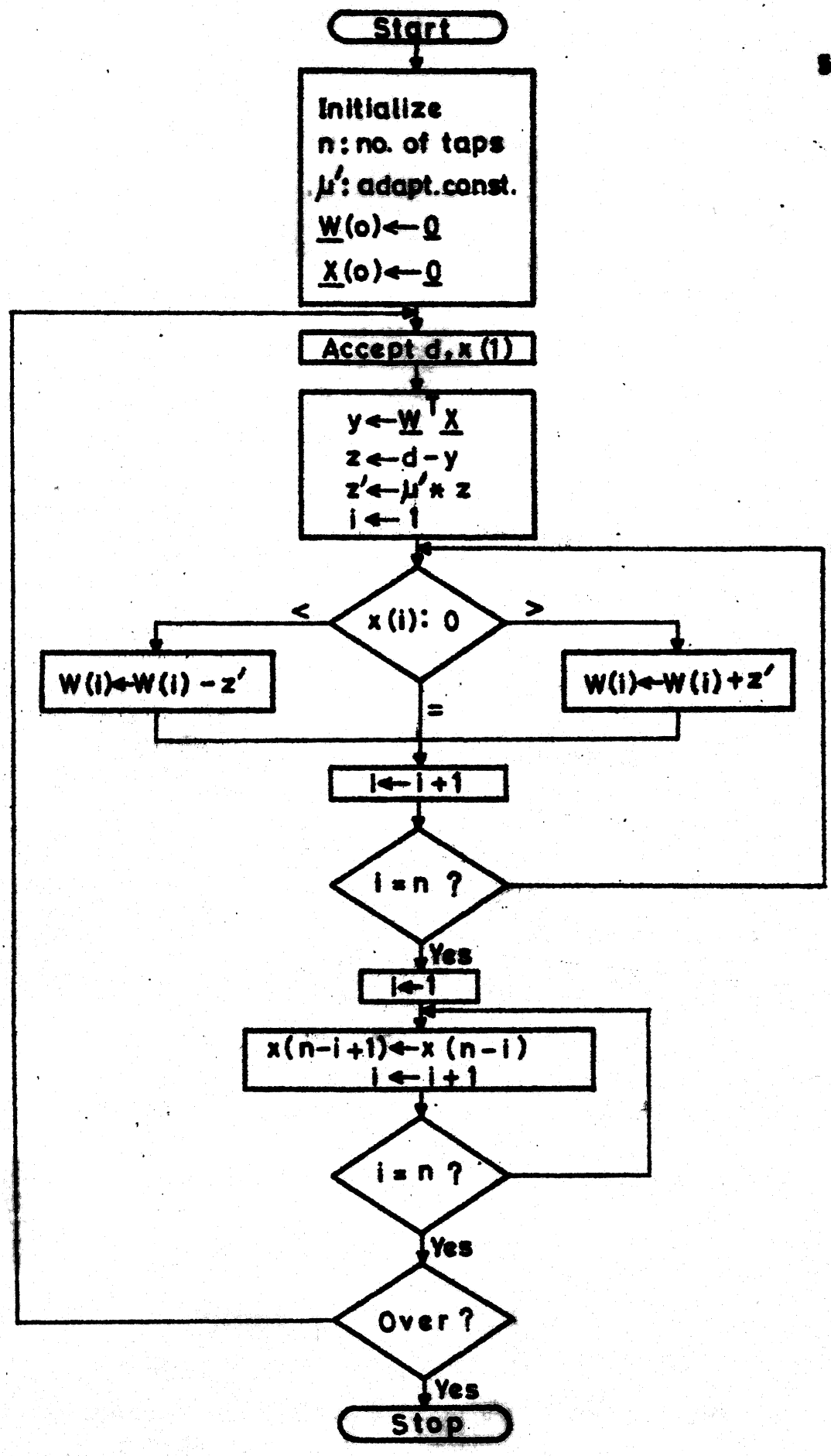


Fig. 5.1: Flowchart for the clipped-data LMS algorithm.

to a coarse quantization when analog signals are processed), the weights by 16 bits and the error by 24 bits. If the constant μ', where $\mu' = 2\mu$, is chosen as a power of 2, then multiplication by μ' is tantamount to a series of shift operations. Accordingly, μ' is chosen as 2^{-14}, and it can be decreased down to 2^{-20} in steps of 2^{-1}. The number of taps can be preset. The specifications of the 8080-processor-based ANC is given in Table 5.2 (page 5-9).

A listing of the assembly language program is given in Appendix A. The following sequence of operations are performed by the program: To begin with the tap weights $\underline{W}$ and the noise samples $\underline{X}$ are initialized to zero; the input data are then accepted, and the convolution summation $\sum_i w(i)\, x(i)$ is formed; the sum is subtracted from the primary input to yield the output z; next, a series of right shift operations are performed on z, and the shifted value is added or subtracted from $w(i)$ depending on sgn $x(i)$; after updating the weights, the next input data are accepted for the subsequent iteration. By suitably manipulating the addresses, the program avoids shifting of reference data samples corresponding to their propagation through the transversal filter.

5.3 Hardware Scheme

With a minor change in the program, the ANC can be implemented as shown in Figure 5.2. The program may be

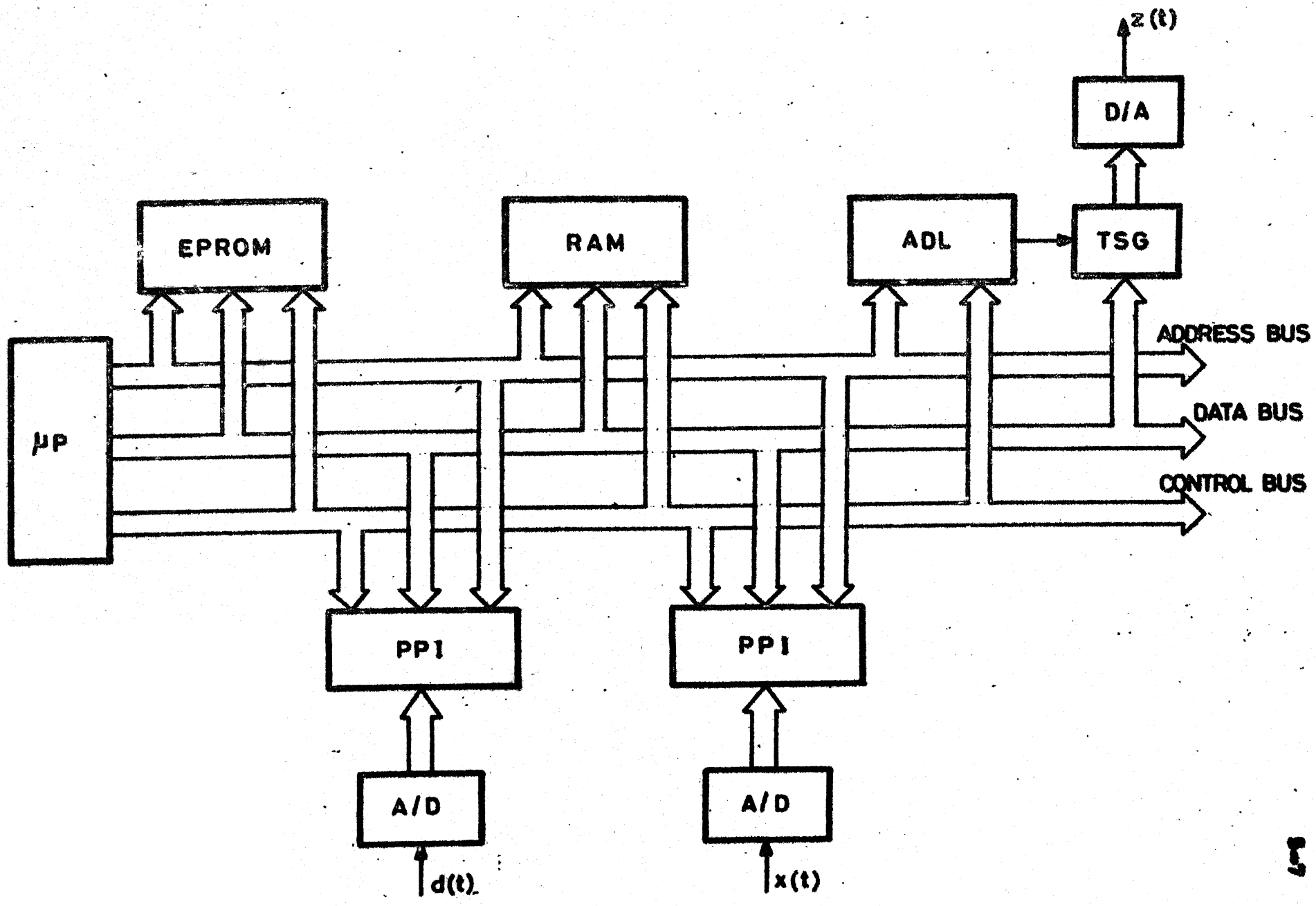


Fig. 5.2: Hardware configuration.

stored in EPROM's and the weight values and the reference samples in RAM's. A programmable peripheral interface (PPI) serves as the interface between the processor and the A/D converter.

For addressing the PPI's in order to read the inputs, memory mapped I/O scheme may be employed, thereby treating I/O devices as memory locations. In 8080, the instruction MOV A,M would transfer one byte of data from a PPI to the accumulator, if its address is stored in the register-pair, HL. Data output could be accomplished by means of an address decoder logic (ADL) and a tri-state gate (TSG).

With the help of a logic state analyzer (Type: HP 1610) timing measurements were made on the developed software routine. The computation times per iteration for different numbers of taps are given in Table 5.3. The program segment MULTI takes a maximum of 700 μs. The clock period of the processor was 0.595 μs (minimum clock period = 0.385 μs).

Considerable speed improvement could be achieved by replacing the software multiplier by a hardware parallel multiplier. Also, multiplication by logarithmic table look-up [5-6] and combinatorial methods [5-7] show promise here. A potentially fast implementation may use programmable CCD's and hard-wired logic circuits. The microprocessor based ANC is, of course, suited to only low-frequency signal

Table 5.1 : ANC specifications.

I/O word length	:	9 bits
Adaptation constant	:	$2^{-14} - 2^{-20}$
Number of taps (min.)	:	1
Iteration period (min.)	:	4 ms
Sample rate (max.)	:	250 Hz
Weight vector update rate (max.)	:	250 Hz
Arithmetic	:	Fixed point

Table 5.2 : ANC computation time.

No. of taps	Computation time/iteration (ms)
2	3.790
5	7.298
16	21.790
32	48.820

extraction. With a slight modification in the program, 16-bit data can be processed. However, it may be preferable to employ 16-bit processors.

SECTION 6

CONCLUSIONS

6.1 Summary

The concept of adaptation in noise cancelling has proven to be a powerful and versatile means of signal processing where precise a priori filter design is impractical. As was seen, no a priori knowledge of signal and noise characteristics is required; pertinent knowledge of their characteristics is acquired through a process of adaptation. Implementation of noise cancellers in their most general form [6-1] is, of course, beset with severe practical constraints in that they are computationally intractable; however, with the approximation introduced by the LMS algorithm, their realization turns out to be considerably simplified.

For the LMS algorithm, simplified design equations, which are at present available in a scattered form in the literature, have been presented in a unified manner. Towards this unification, the main features of the treatment presented here are the following:

1) Derivation, from first principles, of the expression for the time constants of the weight vector.

2) Derivation of the expressions for rate of adaptation and condition for convergence for the

n-dimensional case of the CLMS algorithm.

3) Relative evaluation of the computations involved in the various adaptation algorithms considered in this work.

4) A new method for reduction of computation and its applications to some of the existing algorithms.

5) Providing quantitative arguments for some of the explanations given in the literature on qualitative lines.

6) Examining the possibility of implementing real-time noise cancellers by means of microprocessors.

Besides the LMS algorithm, some of the recent algorithms were also discussed and compared in the work presented here.

In addition to theoretical analysis, simulation studies play an important role in systems such as ANC's. Extensive simulations were carried out bearing on different aspects, some of them duplicating previous work and some designed specifically in the course of this work. On the whole, the results of simulation presented here should serve to provide a clear understanding of the basic principles and also reveal some of the finer features of ANC's.

A quantitative analysis on the advantages and limitations of noise cancellers was included to indicate the level of performance attainable in practical environments. As for their realization, the microprocessor based ANC constitutes a practical possibility for low frequency applications. Moreover, with a digital processor to implement the ANC's, it may also be possible to program

the design rules for the optimal choice of parameters; in that case the ANC's may truly be said to be "self-designing".

6.2 Scope for Further Work

The applications of ANC's to areas wherein fixed filters are now employed may be systematically explored, as this would ease the burden of stringent prior design on the part of the designer. Besides, an attempt may be made to study and compare the various properties of the numerous adaptive algorithms available in the literature. Also, the MLMS algorithm, which is otherwise superior to the other algorithms, needs further attention in respect of its simplified realization. Finally, it may be mentioned that, as pointed out in [2-2], there is considerable scope for theoretical research in what has been called "the statistical theory of adaptation".